



Conception et déploiement des Systèmes de Production Reconfigurables et Agiles (SPRA)

Imad Chalfoun

► To cite this version:

Imad Chalfoun. Conception et déploiement des Systèmes de Production Reconfigurables et Agiles (SPRA). Micro et nanotechnologies/Microélectronique. Université Blaise Pascal - Clermont-Ferrand II, 2014. Français. NNT : 2014CLF22488 . tel-01247600

HAL Id: tel-01247600

<https://theses.hal.science/tel-01247600>

Submitted on 4 Jan 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre : D. U : 2488

EDSPIC : 667

UNIVERSITÉ BLAISE PASCAL – CLERMONT II

ÉCOLE DOCTORALE
SCIENCES POUR L'INGÉNIEUR DE CLERMONT-FERRAND

THÈSE DE DOCTORAT

Présentée par

Imad CHALFOUN

Pour obtenir le grade de

DOCTEUR D'UNIVERSITÉ

SPÉCIALITÉ : Génie Mécanique

(Orientation : Génie Industriel)

Conception et déploiement des Systèmes de Production Reconfigurables et Agiles (SPRA)

Soutenue publiquement le 26 Septembre 2014 devant le jury :

M. Pierre CASTAGNA, Professeur, Université de Nantes.

Président et Rapporteur

M. Yves SALLEZ, Maître de Conférences HDR, Université de Valenciennes
et du Hainaut-Cambrésis.

Rapporteur

M. Pascal RAY, Professeur, École Nationale Supérieure
des Mines de Saint-Étienne.

Directeur de thèse

M. Khalid KOUISS, Maître de Conférences, IFMA de Clermont-Ferrand.

Co-encadrant

M. Arnaud LELEVÉ, Maître de Conférences, INSA de Lyon.

Invité

M. Nicolas BOUTON, Maître de Conférences, IFMA de Clermont-Ferrand.

Invité

Institut Pascal de Clermont-Ferrand (UMR 6602 CNRS/UBP/IFMA)

Mémoire de thèse

Conception et déploiement des Systèmes de Production Reconfigurables et Agiles (SPRA)



Imad CHALFOUN

Dédicace

À l'âme de mon père qui m'a aidé, encouragé, conseillé, guidé et soutenu jusqu'au dernier moment de sa vie et j'aurais aimé qu'il soit parmi nous aujourd'hui.

À ma très chère mère...

À mon amour et à ma chère épouse Nagham...

À mes enfants Wissam, Nasra et Céline qui sont les oiseaux et les anges du paradis et ils rendent ma vie heureuse...

À mon frère Rami et sa famille...

À mes sœurs Rania, Nissrine, Haifa, Zeifa et leurs familles...

À mes proches...

À mes amis...

À mes collègues ...

À mes professeurs...

Et à mon cher pays « Syrie ».

Imad

Remerciements

Je tiens à remercier tout d'abord tous les membres du jury pour le temps précieux qu'ils ont consacré à cette tâche et également pour avoir accepté d'évaluer mes travaux de thèse. Je les remercie de plus de leurs remarques positives et constructives.

Mes remerciements s'adressent très vivement à Pr. Pascal RAY, directeur de thèse, à M. Khalid KOUISS et M. Nicolas BOUTON, co-encadrants de thèse, pour m'avoir guidé, encouragé, conseillé, aidé pendant ma thèse, et également pour leur gentillesse.

Egalement, j'adresse mes remerciements à Pr. Pierre CASTAGNA qui a présidé mon jury de thèse et également qui a accepté de rapporter mon travail.

Mes remerciements vont aussi à M. Yves SALLEZ qui a rapporté mes travaux de thèse et de faire partie de mon jury de thèse.

J'exprime toute ma reconnaissance pour l'honneur qu'ils m'ont fait en acceptant de rapporter cette thèse, et également pour leurs nombreuses remarques et suggestions qui ont contribué à améliorer la qualité de mon mémoire de thèse.

Je tiens à remercier également M. Arnaud LELEVE pour avoir accepté d'examiner mon travail et de faire partie de mon jury de thèse.

Je remercie très chaleureusement ma chère mère qui m'a aidé, soutenu, encouragé et conseillé jusqu'aujourd'hui et également qui a accepté de venir de Syrie pour assister à cet événement intéressant pour nous.

Un merci coloré et parfumé à mon amour et mon épouse Nagham pour m'avoir aidé, soutenu, encouragé, et également pour avoir partagé les différentes difficultés pendant ces années de thèse. Je n'oublie jamais nos enfants Wissam, Nasra et Céline qui rendent notre vie heureuse.

Un grand merci pour mon frère, mes sœurs et mes proches qui m'ont encouragé et soutenu tout au long de mes études.

Mes remerciements s'adressent à tous mes amis et mes collègues en Syrie et en France, et à tous les chercheurs et les personnels du laboratoire Institut Pascal et de l'IFMA.

Mes remerciements vont également à l'ISSAT (Institut Supérieur des Sciences Appliquées et de Technologie) à Damas qui m'a permis de continuer mes études universitaires et supérieures en Syrie et en France.

Enfin, mes remerciements ne seraient pas complets sans mentionner la France qui m'a accueilli et m'a permis de continuer mes études supérieures (le master et le doctorat) dans ses universités et dans ses écoles d'ingénieurs très connues dans le monde entier.

Clermont-Ferrand, le 24 octobre 2014

TABLE DES MATIERES

GLOSSAIRE - 15 -

Liste des Acronymes.....- 17 -

Liste des Figures.....- 19 -

Liste des Tableaux.....- 24 -

INTRODUCTION GENERALE - 25 -

PREMIERE PARTIE - 35 -

ETAT DE L'ART

CHAPITRE 1 - 37 -

DES LIGNES FIXES ET DEDIEES VERS L'AGILITE DES SYSTEMES

1. Introduction.....- 39 -

2. Définition d'un système de production.....- 39 -

3. Systèmes de production traditionnels.....- 41 -

3.1. Lignes de production dédiées: (DML: Dedicated Manufacturing Lines).....- 41 -

3.2. Systèmes de production flexibles : (FMS : Flexible Manufacturing Systems).....- 42 -

4. Systèmes de production reconfigurables : (RMS : Reconfigurable Manufacturing Systems).....- 43 -

4.1. Aspects de la reconfiguration.....- 44 -

4.2. Caractéristiques principales de la reconfiguration.....	45 -
5. Comparaisons entre DML, FMS et RMS.....	47 -
6. Aspects de l'agilité.....	51 -
6.1. Métriques d'une réponse efficace.....	52 -
6.2. Structure agile.....	52 -
6.3. Principes fondamentaux de l'agilité.....	54 -
6.4. Systèmes agiles et Lean.....	55 -
7. Conclusion.....	56 -
CHAPITRE 2	- 59 -
CLASSIFICATION DES CONFIGURATIONS D'UN SYSTEME DE PRODUCTION	
1. Introduction	61 -
2. Calcul du nombre total de configurations.....	61 -
3. Configurations symétriques/asymétriques.....	63 -
4. Calcul du nombre de configurations de type RMS.....	66 -
5. Conclusion.....	67 -
CHAPITRE 3	- 69 -
METHODES ET LANGAGES DE MODELISATION D'UN SYSTEME	
1. Introduction	71 -
2. Outils et méthodes traditionnels de modélisation d'un système.....	71 -
3. Le langage UML.....	72 -
3.1. Historique.....	72 -
3.2. Caractéristiques.....	73 -
3.3. Architecture d'UML 2.0.....	74 -

3.4. Extensions et lacunes d'UML.....	- 76 -
4. Le langage SysML.....	- 77 -
4.1. Origines de SysML.....	- 78 -
4.2. Critères de SysML.....	- 79 -
4.3. Présentation synthétique de SysML.....	- 80 -
5. Conclusion.....	- 83 -
CHAPITRE 4	- 85 -
PILOTAGE DES SYSTEMES DE PRODUCTION RECONFIGURABLES	
1. Introduction.....	- 87 -
2. Niveaux de pilotage d'un système de production reconfigurable.....	- 88 -
3. Architectures de pilotage d'un système de production reconfigurable.....	- 94 -
3.1. Architecture de pilotage centralisée.....	- 95 -
3.2. Architecture de pilotage hiérarchisée.....	- 96 -
3.3. Architecture de pilotage hétérarchisée.....	- 97 -
3.4. Architecture de pilotage décentralisée (isoarchique).....	- 98 -
4. Comparaison entre les différentes architectures de pilotage.....	- 103 -
5. Conclusion.....	- 105 -
DEUXIEME PARTIE	- 107 -
CONCEPTION D'UN SYSTEME DE PRODUCTION RECONFIGURABLE ET AGILE (SPRA)	
CHAPITRE 1	- 109 -
PROPOSITION D'UN MODELE GENERIQUE D'UN SPRA	
1. Introduction.....	- 111 -

2. L'approche du Lab-STICC.....	- 112 -
3. Principe d'organisation du modèle générique.....	- 113 -
4. La structure.....	- 115 -
4.1. Formalisation de la structure.....	- 115 -
4.2. Méta-modèle haut niveau de la structure.....	- 116 -
4.3. Représentation de la structure physique.....	- 118 -
4.4. Représentation de la structure logique.....	- 120 -
4.5. Description des connexions.....	- 122 -
5. La configuration.....	- 125 -
5.1. Formalisation de la configuration.....	- 125 -
5.2. Méta-modèle haut niveau de la configuration.....	- 126 -
6. Les opérations.....	- 127 -
7. Conclusion.....	- 129 -
CHAPITRE 2	- 131 -
CARACTERISATION D'UN SPRA	
1. Introduction	- 133 -
2. Etude des exigences.....	- 133 -
2.1. Typologie des exigences.....	- 133 -
2.2. Définition et analyse des exigences du SPRA.....	- 134 -
2.3. Diagrammes des exigences.....	- 136 -
2.4. Allocation des exigences.....	- 139 -
3. Description de haut niveau des fonctionnalités du SPRA.....	- 142 -
4. Description structurelle du SPRA.....	- 147 -
5. Conclusion.....	- 149 -

TROISIEME PARTIE

- 151 -

DEPLOIEMENT D'UN SYSTEME DE PRODUCTION RECONFIGURABLE ET AGILE (SPRA)

1. Introduction.....	153 -
2. Processus de reconfiguration.....	154 -
3. Modélisation des opérations de reconfiguration.....	160 -
3.1. Diagrammes d'activité.....	160 -
3.2. Diagrammes de séquence « système ».....	162 -
4. Architecture de pilotage adaptée au SPRA.....	166 -
5. Fonctionnement opérationnel du SPRA.....	169 -
6. Conclusion.....	175 -

QUATRIEME PARTIE

- 177 -

APPLICATION A UNE LIGNE D'ASSEMBLAGE DE TYPE SPRA

1. Introduction.....	179 -
2. Projet Agil@Prod.....	179 -
3. Présentation des ressources utilisées.....	181 -
3.1. Poste de travail.....	182 -
3.2. Robot mobile AGV.....	184 -
4. Cas d'étude « assemblage des cartes électroniques ».....	185 -
5. Projection de nos approches.....	191 -
6. Interfaces de reconfiguration.....	204 -
7. Conclusion.....	208 -

CONCLUSION GENERALE ET PERSPECTIVES - 209 -

ANNEXES	- 215 -
Annexe A	- 217 -
Outils et méthodes traditionnels de modélisation d'un système	
Annexe B	- 231 -
Diagrammes d'exigences d'un SPRA	
Annexe C	- 235 -
Diagrammes de cas d'utilisation opérationnels d'un SPRA	
REFERENCES	- 241 -



Glossaire

Liste des Acronymes

AGV: Automated Guided Vehicle.

AIP-PRIMECA : Atelier Inter-Etablissements de Productique et Pôle de Ressources Informatiques pour la MECAnique.

ANSI: American National Standards Institute.

APHA-SPRA: Architecture de Pilotage Hybride Adaptée au SPRA.

APTE: APplication des aux Techniques d'Entreprise.

BDD: Block Definition Diagram.

CIM : Computer Integrated Manufacturing.

CIMOSA: Computer Integrated Manufacturing – Open System Architecture.

CMS : Composants Montés en Surface.

CNC: Computer Numerical Control.

CNRS : Centre National de la Recherche Scientifique.

CTT : Centre de Transfert de Technologie.

DML: Dedicated Manufacturing Lines.

ERP : Enterprise Resource Planning.

FMS: Flexible Manufacturing Systems.

GRAI : Graphes à Résultats et Activités Inter-reliés.

IBD: Internal Block Diagram.

IDEF: Integrated computer aided manufacturing DEFinition language.

IFMA : Institut Français de Mécanique Avancée.

IHM : Interface Homme Machine.

INCOSE: International Council on Systems Engineering.

IP : Institut Pascal.

IS: Ingénierie Système.

ISA: International Society of Automation.

ISSAT : Institut Supérieur de Sciences Appliquées et de Technologie.

Lab-STICC : Laboratoire des Sciences et Techniques de l'Information, de la Communication et de la Connaissance.

LESTER : Laboratoire d'Electronique des Systèmes Temps Réel.

MERISE : Méthode d'Étude et de Réalisation Informatique pour les Systèmes d'Entreprise.

MES : Manufacturing Execution System.

MESA: Manufacturing Enterprise Solutions Association.

OMG: Object Management Group.

OMT: The Object-Modeling Technique.

OOSE: Object-Oriented Software Engineering.

PROSA: Product-Resource-Order-Staff-Architecture.

RAMS: Reconfigurable and Agile Manufacturing Systems.

RFID : Radio Frequency IDentification.

RMS: Reconfigurable Manufacturing Systems.

S95: Standard international pour l'intégration des entreprises et des systèmes de pilotage.

SADT: Structured Analysis and Design Technics.

SART: Structured Analysis for Real Time.

SQL : Structured Query Language.

SPRA : Système de Production Reconfigurable et Agile.

SysML: Systems Modeling Language.

UBP : Université Blaise Pascal.

UC: Use Case.

UML: Unified Modeling Language.

UMR: Unité Mixte de Recherche.

Liste des Figures

Introduction générale

Figure 1. Entreprise stable.....	- 28 -
Figure 2. Objectifs de chaque usine.....	- 29 -
Figure 3. Cadre des travaux de thèse.....	- 33 -
Figure 4. Organisation des parties du mémoire de thèse.....	- 34 -

Partie 1

Chapitre 1

Figure 1.1.1. Décomposition d'un système de production.....	- 40 -
Figure 1.1.2. Ligne de production dédiée (DML).....	- 41 -
Figure 1.1.3. Système de production flexible (FMS).....	- 42 -
Figure 1.1.4. Système de production reconfigurable (RMS).....	- 43 -
Figure 1.1.5. Aspects de la reconfiguration d'un RMS.....	- 44 -
Figure 1.1.6. Exemple des outils dédiés, flexibles et reconfigurables.....	- 47 -
Figure 1.1.7. Capacité/Fonctionnalité pour les DML, FMS et RMS.....	- 49 -
Figure 1.1.8. Capacité/Coût pour les DML, FMS et RMS.....	- 50 -
Figure 1.1.9. Modèle d'une structure agile.....	- 54 -
Figure 1.1.10. Systèmes agiles et Lean.....	- 56 -

Chapitre 2

Figure 1.2.1. Nombre total de configurations en fonction du nombre des machines.....	- 62 -
Figure 1.2.2. Quelques configurations pour 5 machines.....	- 63 -
Figure 1.2.3. Deux classes des configurations asymétriques.....	- 64 -
Figure 1.2.4. Trois classes des configurations symétriques.....	- 65 -
Figure 1.2.5. Système RMS pratique.....	- 66 -

Chapitre 3

Figure 1.3.1. Elaboration d'UML.....- 73 -

Figure 1.3.2. Relation entre SysML et UML 2.....- 79 -

Figure 1.3.3. Diagrammes de SysML.....- 82 -

Chapitre 4

Figure 1.4.1. Pyramide du CIM.....- 89 -

Figure 1.4.2. Niveaux de décision dans l'entreprise.....- 92 -

Figure 1.4.3. La modélisation opérationnelle de la gestion des opérations de la production (Norme S95).....- 93 -

Figure 1.4.4. Architectures classiques de pilotage.....- 94 -

Figure 1.4.5. Evolution des architectures opérationnelles de pilotage.....- 94 -

Figure 1.4.6. Architecture de pilotage centralisée.....- 95 -

Figure 1.4.7. Architecture de pilotage hiérarchisée.....- 96 -

Figure 1.4.8. Architecture de pilotage hétérarchisée.....- 97 -

Figure 1.4.9. Architecture de pilotage décentralisée (isoarchique).....- 98 -

Partie 2

Chapitre 1

Figure 2.1.1. Organisation précédente du modèle.....- 112 -

Figure 2.1.2. Principe d'organisation du modèle générique d'un SPRA.....- 114 -

Figure 2.1.3. Méta-modèle haut niveau de la structure.....- 117 -

Figure 2.1.4. Caractéristiques d'un bloc.....- 118 -

Figure 2.1.5. Représentation de la structure physique.....- 119 -

Figure 2.1.6. Représentation de la structure logique.....- 121 -

Figure 2.1.7. Représentation générale d'une activité d'une gamme logique.....- 122 -

Figure 2.1.8. Exemple d'une structure avec les connexions.....- 124 -

Figure 2.1.9. Méta-modèle haut niveau de la configuration.....- 126 -

Figure 2.1.10. Représentation des opérations.....	- 128 -
--	---------

Chapitre 2

Figure 2.2.1. Typologie des exigences.....	- 134 -
---	---------

Figure 2.2.2. Diagramme détaillé des exigences du configurateur.....	- 138 -
---	---------

Figure 2.2.3. Diagramme de cas d'utilisation d'un SPRA.....	- 144 -
--	---------

Figure 2.2.4. Diagramme de cas d'utilisation opérationnels du configurateur.....	- 146 -
---	---------

Figure 2.2.5. Vue globale de l'atelier.....	- 148 -
--	---------

Figure 2.2.6. Contexte organique d'un SPRA.....	- 149 -
--	---------

Partie 3

Figure 3.1. Principe général de reconfiguration du système.....	- 156 -
--	---------

Figure 3.2. Représentation fonctionnelle dynamique de la reconfiguration du système...	- 157 -
---	---------

Figure 3.3. Processus de conception et implantation d'une nouvelle configuration.....	- 159 -
--	---------

Figure 3.4. Intégration du processus de reconfiguration avec le modèle générique.....	- 159 -
--	---------

Figure 3.5. Diagramme d'activité illustrant la recherche d'une configuration.....	- 161 -
--	---------

Figure 3.6. Diagramme d'activité illustrant la conception d'une configuration.....	- 161 -
---	---------

Figure 3.7. Diagramme d'activité illustrant l'implantation d'une configuration.....	- 162 -
--	---------

Figure 3.8. Diagramme d'activité illustrant l'affectation d'une configuration.....	- 162 -
---	---------

Figure 3.9. Diagramme de séquence lors de recherche d'une configuration.....	- 164 -
---	---------

Figure 3.10. Diagramme de séquence lors d'implantation d'une configuration.....	- 165 -
--	---------

Figure 3.11. Proposition d'une nouvelle architecture de pilotage (APHA-SPRA).....	- 167 -
--	---------

Figure 3.12. Architecture globale de commande.....	- 168 -
---	---------

Figure 3.13. Principe général des différentes interactions dans l'entreprise.....	- 169 -
--	---------

Figure 3.14. Décomposition d'un système de production en deux sous-systèmes.....	- 170 -
---	---------

Figure 3.15. Proposition d'une architecture de fonctionnement opérationnel d'un SPRA.....	- 171 -
--	---------

Partie 4

Figure 4.1. Atelier AIP-PRIMECA Auvergne de l'IFMA.....	- 180 -
Figure 4.2. Dimensions industrielles et pédagogiques du projet Agil@Prod.....	- 181 -
Figure 4.3. Automate et écran tactile Pro-face.....	- 182 -
Figure 4.4. Principe de la technologie RFID.....	- 183 -
Figure 4.5. Poste de travail.....	- 184 -
Figure 4.6. Robot mobile AGV.....	- 185 -
Figure 4.7. Exemple de carte électronique.....	- 186 -
Figure 4.8. Configuration de l'atelier de deux lignes d'assemblage (A et B).....	- 188 -
Figure 4.9. Nouvelle configuration de l'atelier de deux lignes d'assemblage (A et B)...	- 190 -
Figure 4.10. Nouvelle configuration de l'atelier de deux lignes d'assemblage (A et C)..	- 190 -
Figure 4.11. Compétences d'une entreprise.....	- 192 -
Figure 4.12. Structure physique de la ligne d'assemblage : cas général.....	- 193 -
Figure 4.13. Composants de la structure physique-cas d'étude.....	- 194 -
Figure 4.14. Composants de la structure logique-cas d'étude.....	- 196 -
Figure 4.15. Composants des opérations-cas d'étude.....	- 199 -
Figure 4.16. Composants de la configuration physique-cas d'étude.....	- 200 -
Figure 4.17. Composants de la configuration logique-cas d'étude.....	- 200 -
Figure 4.18. Configuration 1 de cette ligne d'assemblage.....	- 201 -
Figure 4.19. Méthodologie de la reconfiguration-cas d'étude.....	- 202 -
Figure 4.20. Configuration 2 de cette ligne d'assemblage.....	- 203 -
Figure 4.21. Interface principale de reconfiguration.....	- 204 -
Figure 4.22. Satisfaction des exigences par un bloc (interfaces)-cas d'étude.....	- 205 -
Figure 4.23. Sous interface «Saisir une nouvelle configuration».....	- 206 -
Figure 4.24. Interface Pro-face pour l'opérateur.....	- 207 -
Figure 4.25. Interface de supervision.....	- 208 -

Annexes

Annexe A

Figure A.1. Outil méthodologique APTE « Bête à cornes ».....	218 -
Figure A.2. Modèle IDEF0 : Développement d'un système.....	219 -
Figure A.3. La méthode MERISE : analyse des données et traitements à chaque niveau.....	220 -
Figure A.4. Grille GRAI.....	221 -
Figure A.5. Matrice des modèles composant CIMOSA.....	222 -
Figure A.6. Exemple de réseau de Pétri.....	223 -
Figure A.7. Exemple : Datagrammes et Actigrammes de SADT.....	225 -
Figure A.8. Organisation générale de la méthode SA-RT.....	226 -
Figure A.9. Diagramme d'interaction de la méthode BOOCH.....	227 -
Figure A.10. Diagramme d'objet de la méthode OMT.....	228 -
Figure A.11. Phases de développement de la méthode OOSE.....	229 -

Annexe B

Figure B.1. Diagramme des exigences du configurateur.....	231 -
Figure B.2. Diagramme des exigences du chef de l'atelier.....	232 -
Figure B.3. Diagramme des exigences d'opérateurs de maintenance.....	233 -
Figure B.4. Diagramme des exigences d'opérateurs.....	234 -

Annexe C

Figure C.1. Diagramme de cas d'utilisation d'un SPRA.....	235 -
Figure C.2. Diagramme de cas d'utilisation opérationnels du configurateur.....	236 -
Figure C.3. Diagramme de cas d'utilisation opérationnels du chef de l'atelier.....	237 -
Figure C.4. Diagramme de cas d'utilisation opérationnels d'opérateurs.....	238 -
Figure C.5. Diagramme de cas d'utilisation opérationnels d'opérateurs de la maintenance.....	239 -

Liste des Tableaux

Partie 1

Chapitre 1

Tableau 1.1.1. Compte rendu sur les caractéristiques principales d'un RMS.....- 46 -

Tableau 1.1.2. Avantages et limitations/inconvénients des DML, FMS et RMS.....- 48 -

Tableau 1.1.3. Comparaison entre les DML, FMS et RMS.....- 49 -

Tableau 1.1.4. Tableau récapitulatif des trois systèmes de production DML, FMS et RMS.....- 51 -

Chapitre 2

Tableau 1.2.1. Comparaison entre le nombre total des configurations possible et le nombre de configurations de type RMS pour un nombre donné de machines.....- 67 -

Chapitre 4

Tableau 1.4.1. Tableau récapitulatif des différentes architectures de pilotage.....- 104 -

Partie 2

Chapitre 2

Tableau 2.2.1. Ensemble des exigences d'un SPRA.....- 136 -

Tableau 2.2.2. Extrait d'allocation des exigences/Chef de l'atelier : « Traces To » et « Refined By ».....- 139 -

Tableau 2.2.3. Extrait d'allocation des exigences/Configurateur: « Traces To », « Containing Class », « Derived From », « Refined By » et « Satisfied By ».....- 140 -

Tableau 2.2.4. Extrait d'allocation des exigences/Opérateurs: « Traces To », « Refined By » et « Satisfied By ».....- 141 -

Tableau 2.2.5. Extrait d'allocation des exigences/Opérateurs de la Maintenance: « Traces To », « Derived From » et « Satisfied By ».....- 141 -

Partie 4

Tableau 4.1. Gammes d'assemblage des cartes électroniques de type A et B.....- 188 -

Tableau 4.2. Gamme d'assemblage des cartes électroniques de type C.....- 189 -

Introduction générale

L'industrie est aujourd'hui, comme elle a toujours été, une pierre angulaire de l'économie pour chaque pays développé. Avoir une base solide en entreprises industrielles est important pour tous les pays car elles poussent et stimulent tous les autres secteurs de l'économie. Les entreprises industrielles offrent également une grande variété d'emplois qui apporte des bonnes conditions de vie dans de nombreux secteurs de la société. En d'autres termes, l'industrie a été et restera la fondation d'une économie forte et aucun autre secteur ne peut la remplacer.

Pourtant, à l'aube du 21ème siècle, l'environnement de production dans le monde entier fait face à de nombreux défis et changements. L'évolution rapide des marchés est de plus en plus fréquente et imprévisible à cause de la concurrence mondiale. Ainsi, l'introduction rapide de nouveaux produits, les changements brusques et radicaux du marché sont à prendre en compte pour répondre aux besoins du client.

Avant les années 70, la demande du marché en biens manufacturés était très élevée et tout ce qui était produit trouvait immédiatement un acheteur parce que les exigences du client étaient simples. Aujourd'hui, le marché est complètement changeant et il faut que les produits soient personnalisables, variés et peu chers pour séduire le client. La concurrence est devenue mondiale avec l'arrivée des productions de pays émergents très compétitifs et très réactifs (**Koren, 2010a**).

La production est donc entrée dans une époque où toutes les entreprises de production doivent faire face à la concurrence dans une économie mondialisée. Cette concurrence mondiale augmente le pouvoir d'achat des clients, ce qui, à son tour, entraîne une introduction fréquente de nouveaux produits à cause de grandes fluctuations dans la demande.

Par conséquent, pour qu'une entreprise soit compétitive et rentable, elle doit savoir rester stable dans un contexte international changeant et elle doit gagner en réactivité et flexibilité afin d'être en mesure de répondre plus rapidement à de nouveaux types de produits, à de nouveaux marchés ainsi qu'à un nouvel environnement sociétal. Ainsi, les entreprises ont besoin de rationaliser et maîtriser les différents flux d'informations, d'harmoniser leur système d'information et d'améliorer leur rentabilité et leur compétitivité en développant des outils de production capables de s'adapter aux modifications de l'environnement avec une bonne productivité et faible coût. Pour répondre à ce besoin, certaines entreprises ont fait le choix de développer des systèmes de production flexibles (FMS), mais ces systèmes n'ont pas été largement adoptés à cause de son coût très élevé. Ainsi, les entreprises cherchent à mettre en œuvre des nouveaux systèmes de production qui produisent non seulement des produits de haute qualité à faible coût, mais aussi qui sont capables de réagir efficacement avec les changements violents de leur environnement tout en continuant de fonctionner malgré la défaillance d'un composant. De plus, la mondialisation provoque la réduction de la durée de vie commerciale d'un produit et induit aussi l'introduction de produits concurrents de plus en plus vite. Les systèmes de production traditionnels sont donc de moins en moins adaptés à ces exigences. Finalement, pour qu'une entreprise soit stable de nos

jours, celle-ci doit posséder trois caractéristiques principales représentées sous la forme d'une chaise à trois pieds sur la Figure 1 (Dashchenko, 2006), (Koren, 2010a) :

- ✚ Conception de produits innovants et personnalisés: chaque produit est développé en accord avec les besoins du client et de manière créative ;

- ✚ Fonctionnement des systèmes de production adaptables dont la capacité peut facilement être modifiée pour s'adapter à la demande du marché. Les fonctionnalités du moyen de production peuvent également être adaptées de façon rentable pour introduire de nouveaux produits avec des délais courts ;

- ✚ Développement de modèles commerciaux qui réagissent aux besoins des clients et à l'évolution du marché afin d'améliorer les ventes.

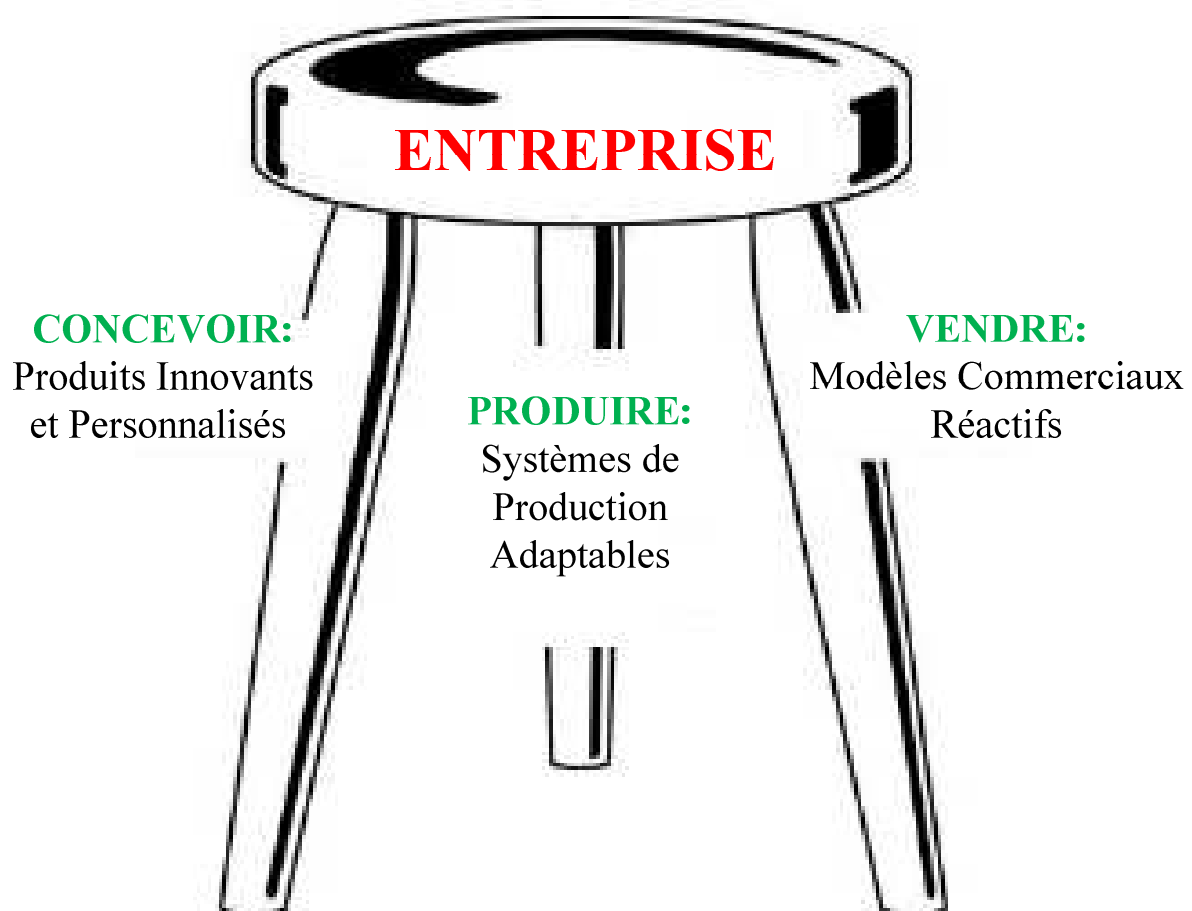


Figure 1. Entreprise stable.

En outre, les entreprises industrielles s'articulent autour de leurs usines qui à leur tour sont au cœur du carré coût-personnalisation de masse-qualité-réactivité. Chaque usine doit à la fois respecter un équilibre entre coût, qualité, produire en série mais sur mesure, et réagir rapidement et de manière rentable aux différents changements. Ainsi, les usines doivent avoir quatre objectifs (Figure 2) :

- ➡ Produire à un coût le plus bas possible : aucune entreprise ne peut survivre longtemps si ses produits coûtent plus cher à produire que ceux de ses concurrents ;
 - ➡ Produire avec une qualité optimale puisque c'est indispensable à la réussite d'un producteur. Sans pour autant faire de la sur-qualité ;
 - ➡ La personnalisation de masse qui se caractérise par une grande variété de produits avec une petite quantité par produit ;
 - ➡ Avoir une grande réactivité, ce qui permet de lancer rapidement de nouveaux produits sur les systèmes de production existants, et de réagir rapidement et de manière rentable (**Dashchenko, 2006**), (**Koren, 2010a**) :
- À l'évolution du marché, y compris les changements dans la demande des produits ;
 - Aux commandes du client: faible coût, haute qualité, produits personnalisés ;
 - Aux changements de produits, y compris les changements dans les produits actuels et l'introduction de nouveaux produits ;
 - Aux changements des réglementations gouvernementales : sécurité et environnement ;
 - Aux défaillances du système: maintenir la production malgré des pannes d'équipements ;
 - Aux changements dans la technologie des procédés.

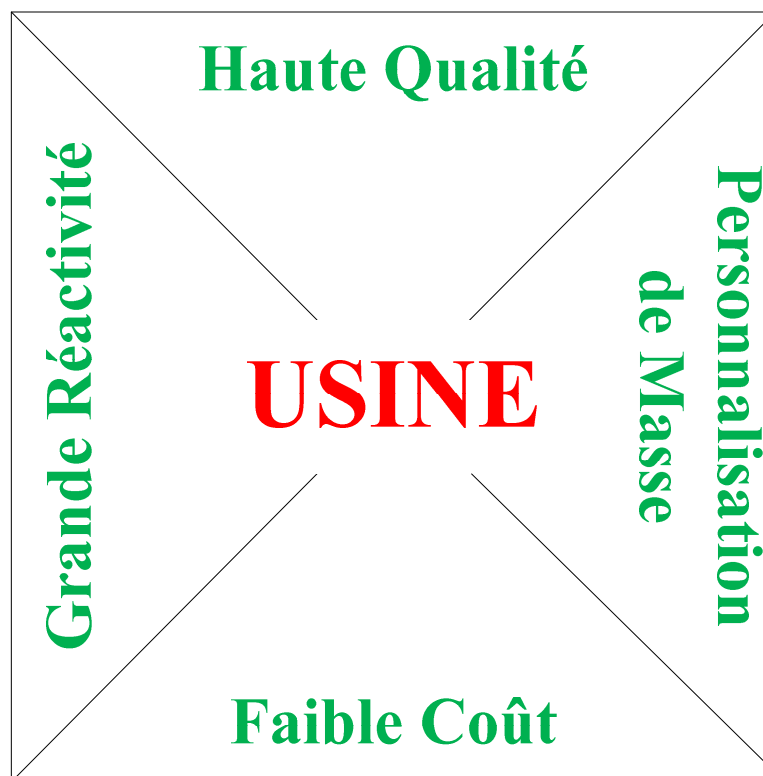


Figure 2. Objectifs de chaque usine.

Il est important de noter que tous ces changements doivent être réalisés dans le contexte d'une concurrence économique agressive à l'échelle mondiale par des clients exigeants.

Bien que la réactivité n'ait pas encore le même niveau d'importance que le coût et la qualité, son effet devient aujourd'hui indispensable puisqu'elle fournit un avantage compétitif dans une économie mondiale instable dans laquelle les entreprises doivent être en mesure de réagir rapidement aux changements et à moindre coût.

Afin de répondre à ces exigences, deux qualités doivent être détenues par les entreprises :

- ✓ La reconfigurabilité ;
- ✓ L'agilité.

Ces deux notions sont définies telles que :

- **Reconfigurabilité** : le système de production reconfigurable est un système conçu au départ pour réagir aux changements rapides dans la structure, ainsi que dans les composants matériels et logiciels, afin d'ajuster rapidement la capacité et la fonctionnalité de la production en répondant aux changements brusques du marché et/ou aux exigences régulières (Koren *et al.*, 1999), (Malhotra *et al.*, 2009). L'objectif de ce type de systèmes de production peut être résumé par la déclaration : « Exactement la capacité et la fonctionnalité nécessaires, uniquement en cas de besoin » (Dashchenko, 2006), (ElMaraghy, 2006).

- **Agilité** : le terme agile a plusieurs significations selon le contexte de son utilisation. Au niveau de l'industrie, l'agilité est la capacité d'un système à se déployer dans un environnement évolutif et imprévisible en apportant une réponse efficace aux différents défis et exigences au sein d'une mission. Le concept d'agilité permet ainsi de réduire le temps nécessaire pour atteindre le marché avec des produits/services adéquates (Dove, 2001), (Boss *et al.*, 2010), (Dove, 2013b).

L'augmentation de la concurrence mondiale, l'évolution rapide du marché, la nécessité de créer des entreprises stables avec des usines rentables obligent la mise en œuvre d'une démarche globale prenant en compte à la fois les aspects techniques, économiques, logistiques et sociétaux lors de la conception d'un système de production innovant.

Cette thèse s'inscrit dans le développement d'un concept innovant de Systèmes de Production Reconfigurables et Agiles (SPRA) permettant de s'adapter rapidement et efficacement aux exigences imposées du marché, des clients, de la technologie des procédés, de l'environnement et de la société afin que l'entreprise soit dynamique, compétitive et rentable.

Dans ces travaux de thèse, la reconfigurabilité repose sur la conception d'un système de production à partir de composants modulaires réutilisables (composants informatiques, postes de travail, connections,...) qui peuvent être agencés

différemment. L'agilité évoque une capacité de mouvement et de transformation en vue de répondre à des modifications de l'environnement. L'agilité traduit donc la dynamique nécessaire au niveau de l'atelier pour passer rapidement d'une configuration à une autre. En d'autres termes, un système de production est dit reconfigurable et agile s'il est capable de fonctionner d'une manière rentable dans un environnement concurrentiel caractérisé par différents changements dont certains imprévisibles. L'adaptation doit alors se faire en un temps minimum.

Dans ce contexte, nous devons répondre à trois questions :

- ☞ Quand reconfigurer ?
- ☞ Quelle configuration viser ?
- ☞ Comment déployer la configuration ?

Il est indispensable de prendre en compte plusieurs aspects :

- 1- La possibilité d'introduire des nouveaux produits.
- 2- La planification de production : variété de produits, taux de production, maintenance. La reconfiguration du système peut être planifiée.
- 3- La réaction aux pannes et à l'arrêt de la production.

L'ensemble de ces paramètres joue un rôle important dans la prise de décision pour reconfigurer le système de production au même titre que le temps nécessaire pour aboutir à la nouvelle configuration afin de répondre à la question « Quand doit-on reconfigurer un système de production ? ».

Lorsque le processus de reconfiguration est déclenché, il faut chercher une nouvelle configuration. Cette nouvelle configuration doit répondre aux différents besoins. Il est donc important de bien évaluer l'ensemble des configurations déjà conçues et enregistrées dans une bibliothèque afin de choisir la meilleure configuration qui pourrait atteindre l'objectif visé. Si aucune configuration ne convient, le configurateur conçoit une nouvelle configuration qui viendra enrichir la bibliothèque. Il s'agira donc de répondre à la question « Pour aboutir à quelle configuration ? ».

Enfin, pour que le configurateur puisse implanter la nouvelle configuration, il est nécessaire de préciser les ressources capables d'intervenir dans cette configuration, de déplacer les postes de travail, et encore d'activer des modules (outils et/ou dispositifs) sur les ressources utilisées selon les exigences de la configuration afin de réimplanter la commande. Cela répond à la question « Comment on reconfigure un système de production ? ».

A partir de ces questions ouvertes, nous pouvons considérer que **la problématique** de cette thèse est de concevoir et implanter une nouvelle configuration. Dans ce contexte, notre travail insiste sur la nécessité et l'intérêt de construire les systèmes de production à partir de modules élémentaires réutilisables (composants informatiques, postes de travail, connections,...) qui peuvent être

agencés différemment et qui permettent de construire de nouvelles configurations à partir de briques existantes. C'est un atout majeur pour la rentabilité des systèmes car il permet de réduire les investissements et augmenter la durée de vie des équipements.

La partie optimisation qui permet de choisir la configuration optimale est nécessaire pour la reconfiguration mais ne sera pas traitée dans la suite des travaux. Par contre, nous proposerons une modélisation du système de production et du processus de reconfiguration afin de pouvoir ajouter facilement un module d'optimisation, ou d'aide à la décision pour le choix optimal de la configuration cible.

En d'autres termes, la problématique de recherche que nous abordons peut s'exprimer de la façon suivante :

Comment modéliser et représenter un système de production en composants réutilisables afin d'assurer la reconfigurabilité et l'agilité de ce système, et de quelle manière déploie-t-on une nouvelle configuration ?

Ces travaux de thèse touchent plusieurs étapes de reconfiguration du système (Figure 3).

L'objectif de la thèse est de contribuer au développement d'un concept innovant de SPRA. Il s'agit de contribuer à la mise en œuvre d'une modélisation des systèmes SPRA et d'une méthodologie pour leur conception, leur déploiement et leur exploitation en intégrant les différents composants réutilisables afin d'aboutir à la reconfigurabilité et l'agilité d'un système de production en répondant aux différentes exigences imposées (lancement d'un nouveau produit, cas de défaillance/panne d'un poste de travail,...). La supervision et le pilotage du système doit être construits de manière à être capable de prendre en charge les capacités de reconfiguration du système.

Nous préconisons une démarche basée sur les concepts de l'ingénierie des systèmes qui permet de caractériser les besoins de reconfiguration puis de les transformer en exigences qui permettent de guider tous les aspects de conception, de construction et d'exploitation du système. A cette fin, la caractérisation, la modélisation, la représentation et l'implantation doivent être assez génériques et il doit passer par la proposition de passerelles avec les outils informatiques et industriels disponibles.

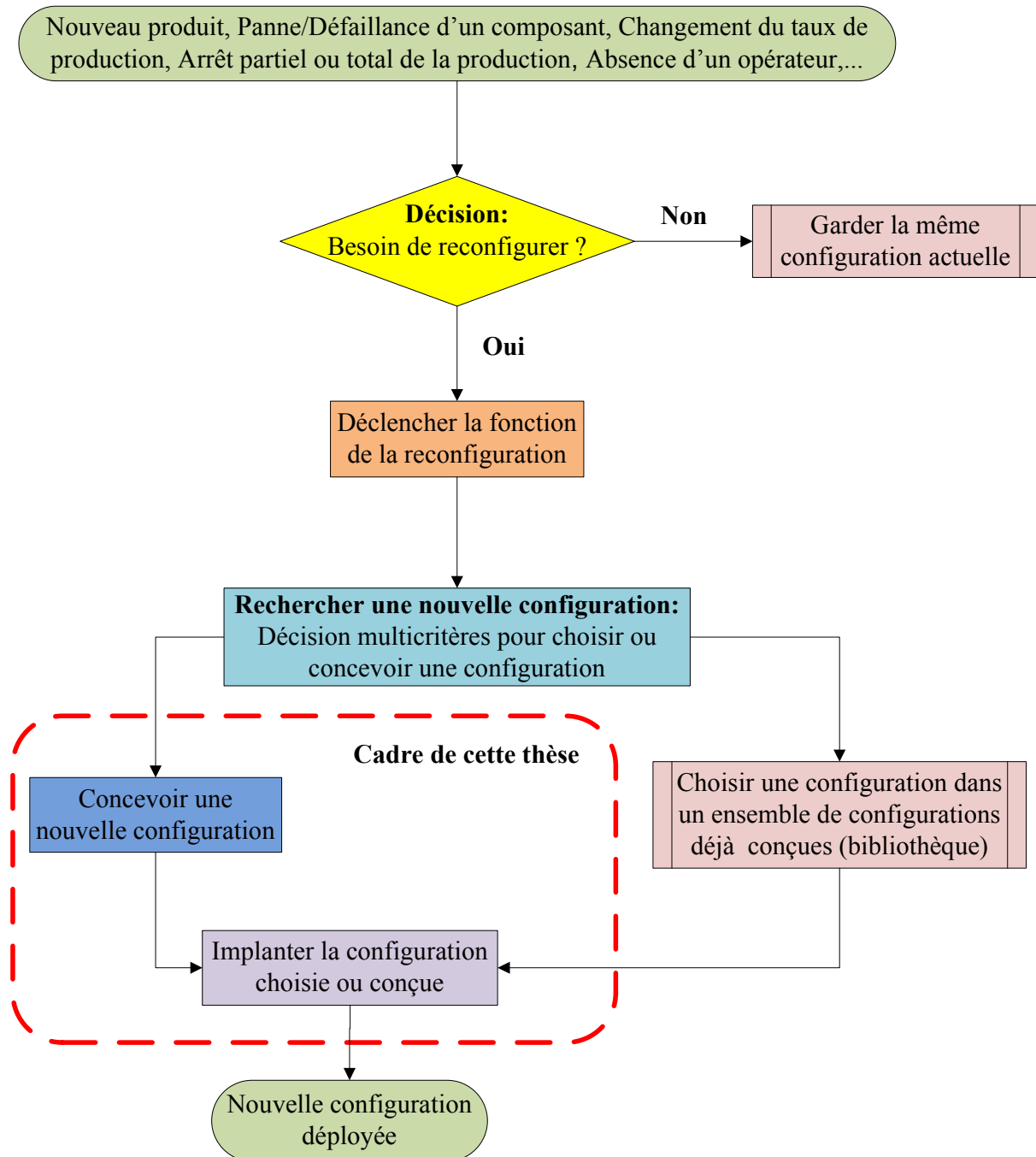


Figure 3. Cadre des travaux de thèse.

Ce mémoire de thèse est organisé en quatre parties (Figure 4). La première partie présente un compte-rendu de l'état de l'art concernant l'évolution des systèmes de production des lignes fixes et dédiées vers des lignes intégrant l'aspect de l'agilité. La classification des différentes configurations possibles d'un système de production sera présentée. De plus, les méthodes et les langages de modélisation d'un système ainsi que les architectures de pilotage d'un système de production reconfigurable seront également présentés dans cette partie.

Afin de répondre à l'objectif de cette thèse, **nos apports** sont présentés dans les trois parties suivantes.

La seconde partie se concentre sur la phase de conception d'un SPRA en utilisant le langage de modélisation des systèmes complexes (SysML : Systems Modeling Language). Un modèle générique adapté à ce type de système sera proposé avec une description et une représentation détaillées de chaque composant du modèle. Ensuite, les exigences permettant de spécifier les capacités d'un SPRA et les contraintes qui doivent être satisfaites par un SPRA seront définies, et également une description de haut niveau des fonctionnalités et de la structure d'un SPRA sera proposée.

La troisième partie répond à la question « comment déploie-t-on une configuration d'un SPRA ? ». Nous proposons une méthodologie à travers un processus de reconfiguration qui représente une démarche à suivre pour concevoir et implanter une nouvelle configuration. De plus, une modélisation des opérations de reconfiguration est proposée sous forme de diagrammes de séquence et de diagrammes d'activité SysML. Une nouvelle Architecture de Pilotage Hybride Adaptée au SPRA (APHA-SPRA) sera proposée. Cette partie introduit également un pilotage opérationnel d'un SPRA.

La dernière partie de ce mémoire est consacrée à l'application du laboratoire à une ligne d'assemblage de type SPRA. Le projet Agil@Prod dans le quel cette thèse s'inscrit sera présenté. Puis, quelques travaux développés au cours de cette thèse seront déployés sur un démonstrateur industriel au sein de la plate-forme AIP-PRIMECA Auvergne qui est une plate-forme de mutualisation dans le domaine de la productique.

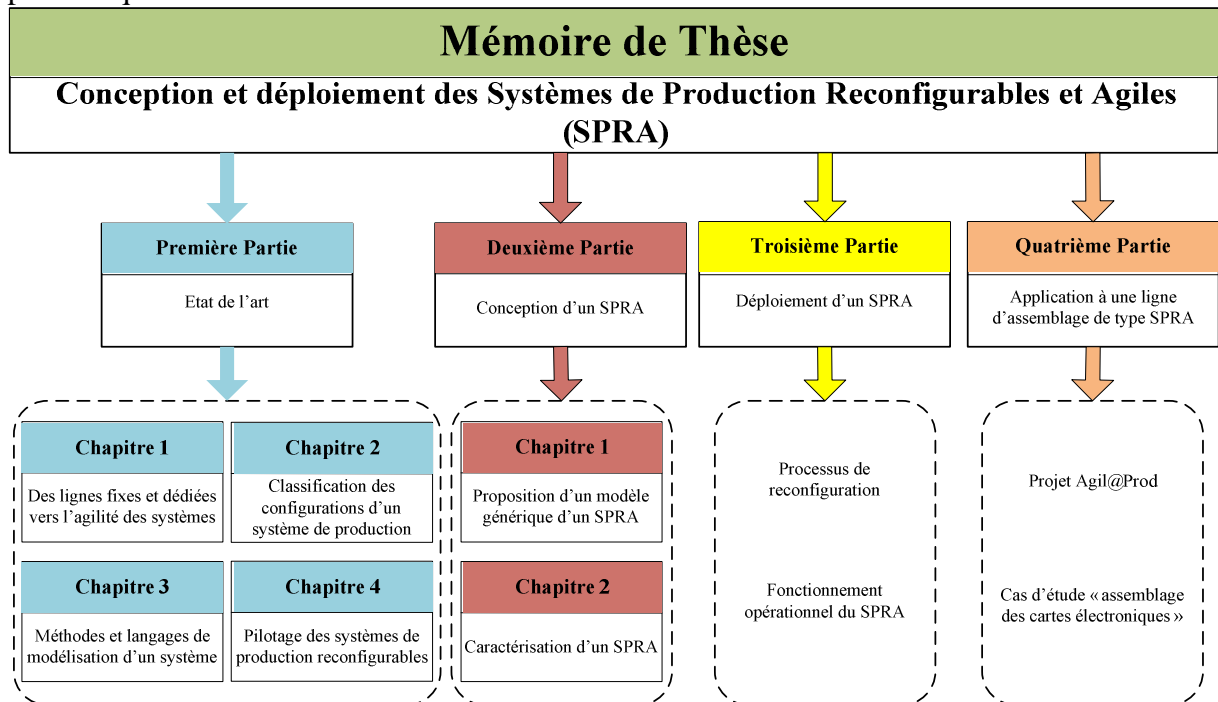


Figure 4. Organisation des parties du mémoire de thèse.

Première Partie

Etat de l'art

Cette partie rend-compte sur l'état de l'art des systèmes de production, les outils de modélisation d'un système et les architecture de pilotage d'un système de production reconfigurable.

Cette partie est constituée de quatre chapitres, le premier chapitre présente l'évolution des systèmes de production des lignes fixes et dédiées vers le système agile. De plus, la classification des différentes configurations possibles d'un système de production sera présentée dans le deuxième chapitre.

Ensuite, le troisième chapitre est consacré à présenter les outils, les méthodes et les langages de modélisation d'un système. Enfin, les architectures de pilotage d'un système de production reconfigurable font l'objet du quatrième chapitre.

Sommaire

Chapitre 1 : Des lignes fixes et dédiées vers l'agilité des systèmes.....	37 -
Chapitre 2 : Classification des configurations d'un système de production.....	59 -
Chapitre 3 : Méthodes et langages de modélisation d'un système.....	69 -
Chapitre 4 : Pilotage des systèmes de production reconfigurables.....	85 -

Chapitre 1

Des lignes fixes et dédiées vers l'agilité des systèmes

L'évolution des systèmes de production à lignes fixes et dédiées vers l'introduction de la notion d'agilité est étudiée dans ce chapitre.

Une définition du système de production sera présentée avant de présenter les systèmes traditionnels (lignes dédiées et systèmes flexibles). Ensuite, nous nous focaliserons sur les systèmes de production reconfigurables. Une analyse et une comparaison entre ces trois types de systèmes de production seront réalisées au travers de l'étude des caractéristiques principales des fonctionnalités, de la capacité et du coût. Enfin, les aspects de l'agilité seront introduit et nous conclurons par quelques remarques issues de cet état de l'art.

Sommaire

1. Introduction	39
2. Définition d'un système de production	39
3. Systèmes de production traditionnels.....	41
3.1. Lignes de production dédiées: (DML: Dedicated Manufacturing Lines)	41
3.2. Systèmes de production flexibles : (FMS : Flexible Manufacturing Systems).....	42
4. Systèmes de production reconfigurables : (RMS : Reconfigurable Manufacturing Systems).....	43
4.1. Aspects de la reconfiguration	44
4.2. Caractéristiques principales de la reconfiguration	45
5. Comparaisons entre DML, FMS et RMS.....	47
6. Aspects de l'agilité	51
6.1. Métriques d'une réponse efficace	52
6.2. Structure agile	52
6.3. Principes fondamentaux de l'agilité.....	54
6.4. Systèmes agiles et Lean.....	55
7. Conclusion	56

1. Introduction

L'environnement de production au 21^{ème} siècle fait face à de nombreux défis et changements. L'évolution des marchés se traduit notamment par l'introduction rapide de nouveaux produits, des changements dans les commandes des clients et de l'évolution des réglementations gouvernementales.

De nos jours, les clients souhaitent que les produits soient personnalisables, variés et peu chers. La concurrence est devenue mondiale avec l'arrivée des productions de pays émergents très compétitifs et très réactifs. Ainsi pour qu'une entreprise soit rentable, elle doit savoir rester compétitive dans un contexte international complexe et mouvant, doit gagner en réactivité et flexibilité et être en mesure de répondre plus rapidement à de nouveaux types de produits, à de nouveaux marchés ainsi qu'à un nouvel environnement sociétal. Elle a besoin de rationaliser et maîtriser les différents flux d'informations, d'harmoniser ses systèmes d'informations et d'améliorer sa rentabilité et sa compétitivité.

La production est donc entrée dans une époque où toutes les entreprises de production doivent se préparer à la concurrence dans une économie mondialisée. Par conséquent, dans la plupart des cas, les systèmes de production traditionnels ne sont plus adaptés à la concurrence sur le marché (**Dashchenko, 2006**), (**Bi et al., 2007**), (**Malhotra et al., 2010**).

Finalement, pour rester compétitives, les entreprises doivent concevoir de nouveaux systèmes de production qui produisent non seulement des produits de haute qualité à faible coût, mais aussi qui permettent de répondre à des changements rapides de leur environnement et de manière efficace (**Bi et al., 2007**), (**Joergensen et al., 2011**).

2. Définition d'un système de production

Un système est un ensemble composite constitué de personnels, de matériels, de logiciels et de procédures. Tous ces éléments sont en interaction mutuelle dans un environnement donné et sont organisés pour répondre à un besoin. Chaque système est déterminé par la nature de ses éléments constitutifs, les interactions entre ces derniers et le critère d'appartenance au système. Par exemple, un système mécatronique est un ensemble complexe et structuré de composants mécaniques, électroniques, automatiques et informatiques en interaction permanente et assurant une fonction d'usage (ex: automobile, avion, train,...). (**NASA, 1995**), (**INCOSE, 2004**), (**AFIS, 2009**).

Nos travaux de recherche s'intéressent au système de production qui transforme les matières premières en produits semis finis en entrant plus tard dans des produits finis répondant aux besoins de clients. Il est constitué de ressources humaines et physiques. En d'autres termes, un système de production peut être

défini comme un ensemble d'opérateurs et de postes de travail et/ou de machines qui sont intégrés pour effectuer une série contrôlée d'opérations répétitives sur les matières premières afin d'obtenir une forme finale souhaitée ou encore pour assembler un ensemble de pièces afin d'obtenir un produit final. Les postes de travail et/ou les machines sont connectés par l'intermédiaire d'un mécanisme de transfert qui peut être un convoyeur, un robot mobile (AGV : Automated Guided Vehicle) ou tout simplement un opérateur.

Par conséquent, les systèmes de production sont considérés comme des systèmes complexes difficiles à gérer au vu de toutes leurs composantes (mécaniques, électroniques, informatiques, maintenance, achats, ventes,...) et du développement continu de la technologie des procédés.

Finalement, afin de mieux comprendre et analyser le fonctionnement des systèmes de production, beaucoup de travaux académiques et industriels ont été publiés (Meinadier, 1998), (Meinadier, 2002), (Le Moigne, 2006). Ce qui a permis de décomposer un système de production en trois sous-systèmes (Figure 1.1.1) :

- Un système d'information : ce sous-système collecte les informations intérieures et extérieures au système, il les stocke et traite et enfin il transmet ces informations différemment et utilement aux autres sous-systèmes.
- Un système de décision : il donne les ordres au système physique prenant en compte les données transmises du système d'information.
- Un système physique : son rôle est de transformer les matières premières en produits à l'aide des ressources humaines ou matérielles (machines, convoyeurs, robots,...).

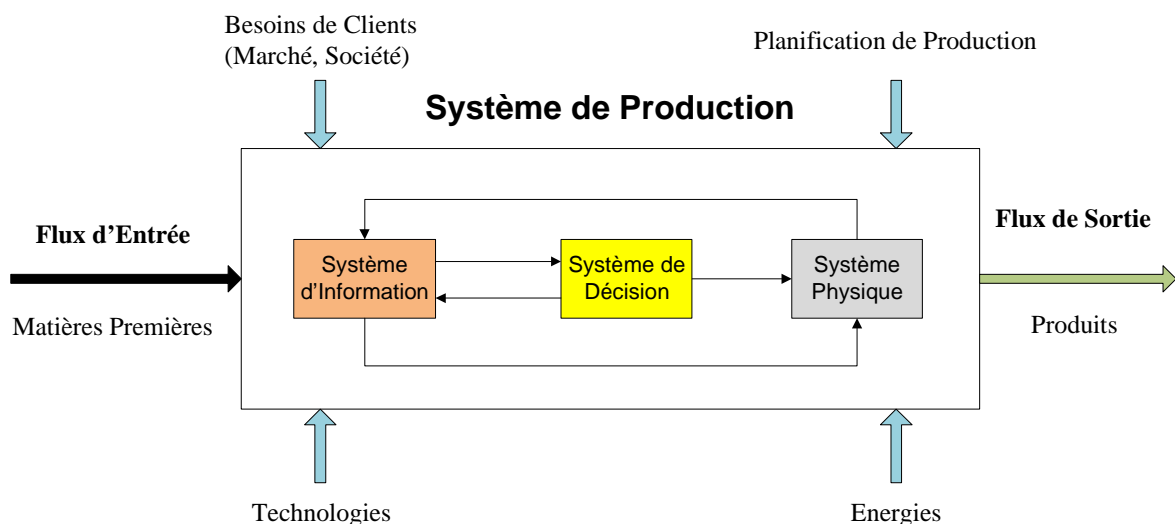


Figure 1.1.1. Décomposition d'un système de production.

Dans le contexte industriel actuel, ces trois sous-systèmes doivent être intégrés d'une manière plus robuste et plus fiable afin d'améliorer continuellement les performances des systèmes de production, et plus précisément leur réactivité et leur productivité ce qui réagit positivement avec le coût et la qualité des produits.

3. *Systèmes de production traditionnels*

A la fin du 20^{ème} siècle, les systèmes de production étaient principalement dominés par deux types : les lignes de production dédiées où les machines sont dédiées à un ensemble fixe d'opérations et les systèmes de production flexibles qui pouvaient produire un ensemble de produits différents car les machines pouvaient changer de tâche comme c'est le cas pour les robots ou les machines à commande numérique (CNC : Computer Numerical Control).

3.1. *Lignes de production dédiées: (DML: Dedicated Manufacturing Lines)*

Les lignes de production dédiées, aussi appelées lignes de transfert, ont été introduites en 1913 (ElMaraghy, 2012). Elles sont basées sur un ensemble de machines relativement simples disposées séquentiellement dans une ligne (Figure 1.1.2). Ces machines sont équipées d'une automatisation fixe adaptée à des produits de base d'une entreprise à haut volume. Chaque ligne dédiée est généralement conçue pour produire un type unique de produits à un haut rythme de productivité. A son tour, chaque machine est destinée à faire, par exemple, une opération d'usinage toujours identique et toujours au même endroit (perçage, fraisage,...). Evidemment, lorsque le volume est élevé, le coût par produit est relativement faible. Les DML sont donc rentables tant que la demande du marché dépasse l'offre car elles travaillent à pleine capacité. (Koren *et al.*, 1999)

Le principal inconvénient des DML est que ces lignes ne sont pas conçues pour être modifiables. Par conséquent, elles ne peuvent pas être converties facilement pour produire de nouveaux produits. Ainsi, leur rôle dans la production moderne est de moins en moins important (Dashchenko, 2006). De plus, il y a beaucoup de situations dans lesquelles les lignes dédiées ne fonctionnent pas à pleine capacité, ce qui génèrent ainsi des pertes.

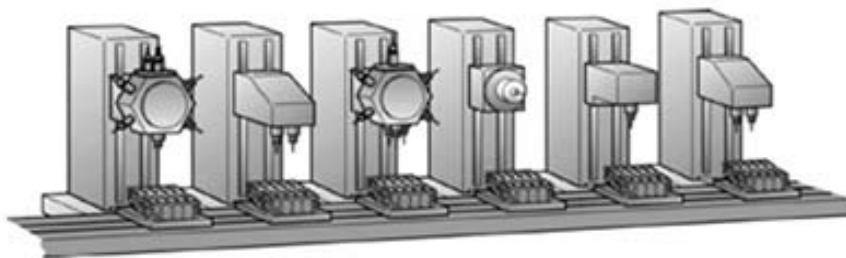


Figure 1.1.2. Ligne de production dédiée (DML).

De nos jours, les DML ne sont donc pas rentable et disparaissent dans de nombreuses entreprises industrielles (**Koren, 2010a**).

Pour remédier à ce genre de problème, certaines entreprises ont fait le choix de s'orienter vers des systèmes de production flexibles.

3.2. Systèmes de production flexibles : (FMS : Flexible Manufacturing Systems)

Ce type de systèmes de production est apparu au début des années 70 (**ElMaraghy, 2012**). Il est composé de machines CNC programmables et de systèmes de pilotage et de supervision, ils coûtent donc plus chers qu'un DML, mais peuvent produire une variété de produits avec un volume variable sur le même système de production. Dans un FMS, il est difficile d'ajouter une nouvelle machine ou de changer une machine. Autrement dit, la structure du FMS est souvent figée (Figure 1.1.3).

De plus, les FMS sont capables de produire n'importe quel produit même ce dont nous n'avons pas besoin et cela nécessite normalement plus de fonctions que nécessaire, tout cela rend le prix d'un coût du système très élevé qui explique plutôt le relatif échec des FMS. C'est pourquoi les FMS n'ont pas été largement adoptés dans les usines, et la plupart des entreprises qui ont un système de production de type FMS ne sont pas satisfaits de leur performance (**Dashchenko, 2006**), (**ElMaraghy, 2006**), (**Malhotra et al., 2009**), (**Koren, 2010a**).

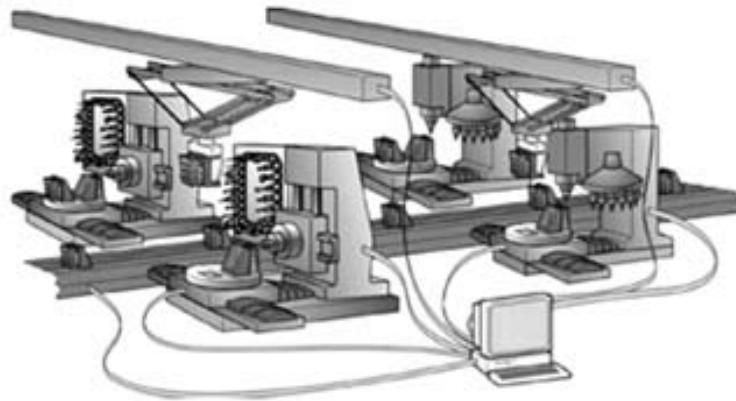


Figure 1.1.3. Système de production flexible (FMS).

Par conséquent, avec la concurrence actuelle des marchés dans une économie mondialisée, les systèmes de productions traditionnels DML et FMS ne sont pas en mesure de répondre aux exigences de réactivité du marché à un coût raisonnable.

Dans ce contexte, une nouvelle classe de systèmes de production a émergé. Elle permet de combiner le haut débit de production des DML avec la flexibilité des FMS, et en plus d'être réactif et capable de réagir efficacement et rapidement aux

différents changements possibles. Ce type de systèmes est appelé les systèmes de production reconfigurables.

4. Systèmes de production reconfigurables : (RMS : Reconfigurable Manufacturing Systems)

Le système de production reconfigurable (RMS) est un système qui fait le pont entre le DML et le FMS. Théoriquement, RMS est né en 1990 dans un environnement concurrentiel caractérisé par différents changements continus et imprévisibles.

Ainsi, le RMS est défini dans (Koren *et al.*, 1999), (Spicer *et al.*, 2005), (Malhotra *et al.*, 2009) et (ElMaraghy, 2011) comme : « un système conçu au départ pour réagir aux changements rapides dans la structure, ainsi que dans les composants matériels et logiciels, afin d'ajuster rapidement la capacité et la fonctionnalité de la production en répondant aux changements brusques du marché et/ou aux exigences régulières ».

Par ailleurs, les RMS sont définis dans (Mehrabi *et al.*, 2000a) comme la clé du futur pour l'industrie. Les auteurs estiment que des systèmes de ce genre doivent avoir une conception rapidement élaborée et une grande facilité à se convertir en fonction de la demande. Il doit aussi être capable d'intégrer de nouvelles technologies et de produire un nombre croissant de variétés de produits en quantités variées.

La Figure 1.1.4 présente un RMS qui permet d'ajuster rapidement la capacité de production en ajoutant de nouvelles machines ou modifiant les machines déjà installées dans les différentes phases, et toutes ces machines peuvent être facilement amovibles. Effectivement, la structure du RMS est modifiable et n'est donc pas figée comme elle est souvent dans le cas du FMS.

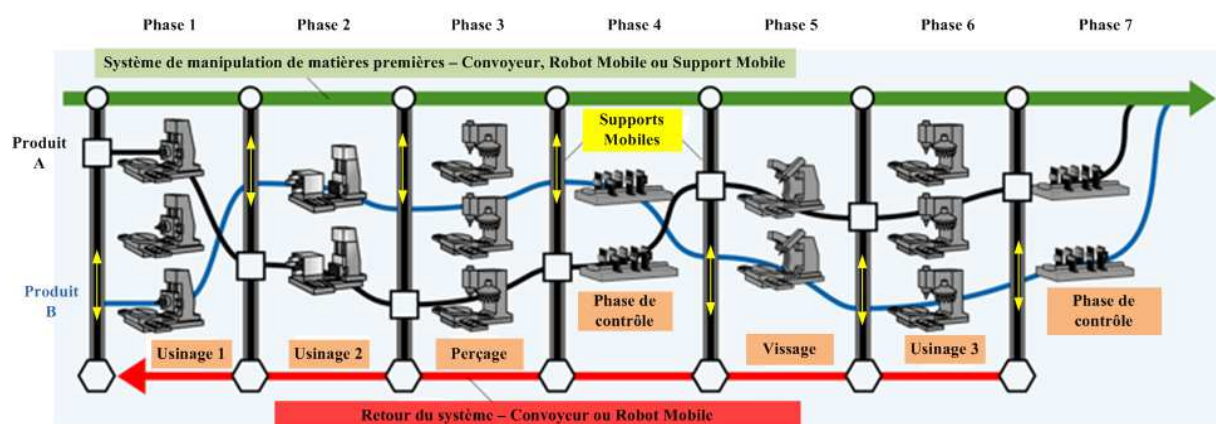


Figure 1.1.4. Système de production reconfigurable (RMS).

L'objectif d'un RMS peut être résumé par la déclaration : « Exactement la capacité et la fonctionnalité nécessaires, uniquement en cas de besoin » (ElMaraghy, 2006).

Afin de réagir à l'évolution permanente et imprévisible de la demande de produits, à la variété des produits et aux technologies de procédés, les RMS permettent un ajustement rapide des capacités de production et de la fonctionnalité des machines en réorganisant ou en modifiant leurs composants modulaires (Bruccoli *et al.*, 2003), (Malhotra *et al.*, 2010).

4.1. Aspects de la reconfiguration

Il s'agit notamment de diverses configurations du système de production. La Figure 1.1.5 montre qu'il y a de nombreux aspects de la reconfiguration, par exemple, la reconfiguration au niveau (Mehrab *et al.*, 2000a), (Mehrab *et al.*, 2000b) :

- ✚ La morphologie de la structure du système : en série, en parallèle ou hybride ;
- ✚ Les logiciels/matériels des communications entre les composants mécaniques, électriques et électroniques doivent être modulaires et reconfigurables pour être capables de s'adapter aux différentes configurations possibles ;
- ✚ Les méthodes de mesure et de contrôle ;
- ✚ Les machines modulaires qui peuvent être créées, selon les besoins, de certains modules dits «composants de base» ;
- ✚ L'outillage modulaire ;
- ✚ Les processus modulaires de réglage et de diagnostique.

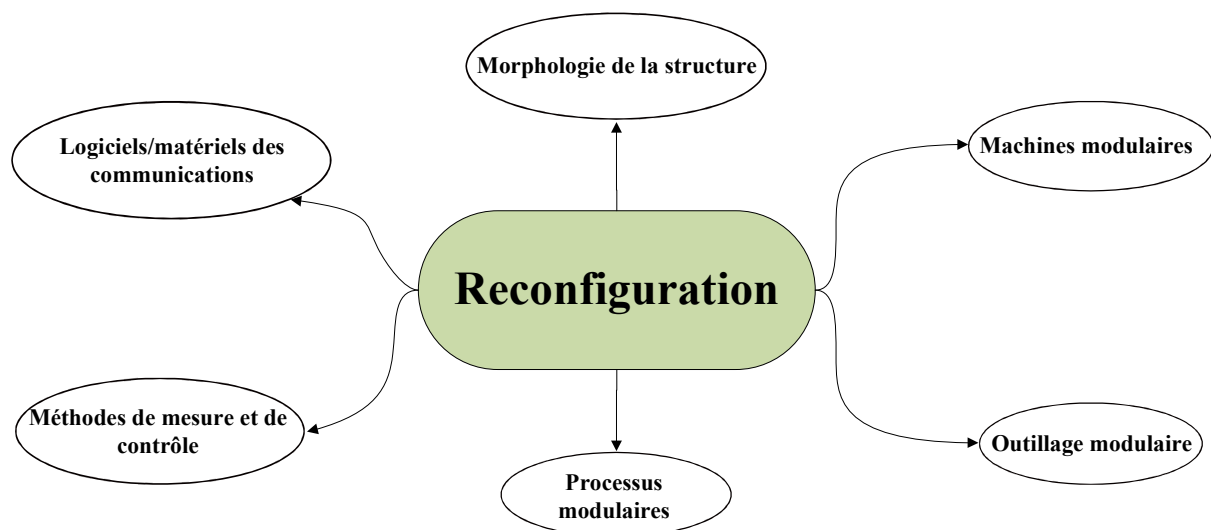


Figure 1.1.5. Aspects de la reconfiguration d'un RMS.

4.2. Caractéristiques principales de la reconfiguration

Les RMS doivent être conçus au départ en utilisant des composants matériels et logiciels qui peuvent être intégrés rapidement et de manière fiable.

Typiquement, selon les auteurs (**Koren *et al.*, 1999**), (**Mehrabi *et al.*, 2000a**), (**Dashchenko, 2006**), (**ElMaraghy, 2006**), chaque RMS doit posséder les six caractéristiques principales suivantes :

1- Modularité : dans un RMS tous les composants principaux doivent être modulaires (les ressources, les outils, le contrôle,...). Lorsque nécessaire, les composants modulaires peuvent être remplacés ou modernisés afin de mieux répondre aux nouvelles exigences et/ou applications. La sélection des modules de base et la façon dont ils sont connectés doivent permettre la création de systèmes qui peuvent être facilement intégrés, diagnostiqués et convertis.

2- Intégrabilité : les machines et les modules doivent être conçus avec des interfaces permettant l'intégration rapide des composants. La performance du système intégré est définie en se basant sur les performances des composants, des logiciels ainsi que des interfaces.

3- Personnalisation : cette caractéristique distingue radicalement le RMS du FMS et de la DML parce qu'elle permet une réduction des coûts du système. Elle touche deux aspects : une flexibilité et un contrôle personnalisés. Le premier aspect permet de signifier que les machines ne fournissent que la souplesse nécessaire pour chaque lot de produits réduisant ainsi les coûts. Le deuxième est obtenu en intégrant des modules de contrôle avec l'aide de la technologie fournissant uniquement les fonctions de contrôles recherchées.

4- Convertibilité : La convertibilité du système est la capacité à changer rapidement et facilement les fonctionnalités du système et ses machines installées afin de s'adapter aux nouveaux besoins de la production. Ainsi, la convertibilité du système inclut la convertibilité des machines qui peut être caractérisée par le changement des outils, une partie des programmes, ou encore un réglage des degrés de liberté.

5- Évolutivité : l'évolutivité du système est une caractéristique qui permet de modifier facilement la capacité de production. L'évolutivité peut être caractérisée par l'ajout de parties à une machine pour augmenter sa productivité, ou encore en additionnant des machines pour augmenter la capacité globale du système de production.

6- Diagnostic : dans un RMS, la détection rapide des sources de défaillance au niveau de la qualité, et de panne des machines permettra de réduire les délais de démarrage. Cette caractéristique « diagnostic » est la capacité de lire automatiquement l'état actuel du système pour détecter et diagnostiquer les causes

fondamentales de défauts des produits, et également pour corriger rapidement les défauts de fonctionnement.

L'expert international des RMS (**Koren, 2010a**) considère que la personnalisation, l'évolutivité et la convertibilité sont les caractéristiques nécessaires pour la reconfiguration. En outre, la modularité, l'intégrabilité et le diagnostic sont les caractéristiques suffisantes pour la reconfiguration.

Ainsi, il existe une bonne relation entre ces six caractéristiques et les objectifs des systèmes de production. Elles améliorent la réactivité et la productivité du système et réduisent également le temps, les efforts et les coûts de la reconfiguration.

En résumé, le Tableau 1.1.1 permet de montrer l'influence de ces six caractéristiques dans le contexte de la reconfigurabilité. La modularité, l'intégrabilité, la convertibilité et le diagnostic réduisent le temps de reconfiguration, et améliorent ainsi la réactivité du système. En dehors de la modularité, ces caractéristiques contribuent à la réduction des coûts du cycle de vie du système en lui permettant de changer de configuration au cours de sa durée de vie, et également de rester compétitif malgré l'évolution des marchés, des demandes de clients et des technologies de procédés. Tout cela réduit les investissements et augmente la durée de vie des équipements. Par ailleurs, un système modulaire peut être plus coûteux en raison du coût supplémentaire des interfaces.

De ce point de vue, les RMS semblent être un choix prometteur pour les entreprises industrielles car ils augmentent effectivement la productivité et la réactivité du système et en même temps ils réduisent les coûts de production.

	Temps de reconfiguration	Productivité	Coûts du cycle de vie du système
Modularité Les composants sont modulaires	↓		
Intégrabilité Des interfaces pour une intégration rapide	↓		↓
Personnalisation La flexibilité nécessaire justement en cas de besoin		↑	↓
Evolutivité La capacité est modifiable		↑	↓
Convertibilité Les fonctionnalités sont modifiables	↓	↑	↓
Diagnostic Le système est « diagnostiquable »	↓	↑	↓

Tableau 1.1.1. Compte rendu sur les caractéristiques principales d'un RMS.

De plus, de nombreux travaux de recherches, (Wiendahl *et al.*, 2004), (Wiendahl *et al.*, 2007), (ElMaraghy, 2011), ont récemment identifié et ajouté un nouvel item à ces six caractéristiques principales. Il s'agit de la mobilité qui peut être définie comme : « Les composants et les sous-systèmes sont déplaçables et implantables facilement et rapidement pour réagir efficacement aux nouveaux besoins ». Actuellement, cette nouvelle caractéristique est un objectif très présent chez beaucoup d'industriels.

5. Comparaisons entre DML, FMS et RMS

Tout d'abord, nous allons prendre un exemple pratique de notre vie quotidienne pour bien expliquer la différence entre les trois termes : « dédiée », « flexible » et « reconfigurable ». Pour l'illustrer, nous pouvons utiliser le concept de clés (Figure 1.1.6) :

- La clé « dédiée » se consacre à une seule taille d'écrous ;
- La clé « flexible » est un outil réglable selon la taille de chaque écrou ;
- La clé « reconfigurable » ou la clé à cartouches modulaires peut tourner les écrous dont la géométrie correspond à une famille d'écrous.

Ainsi, les outils reconfigurables permettent de combler les lacunes entre les outils dédiés et les outils flexibles (Koren, 2010a).

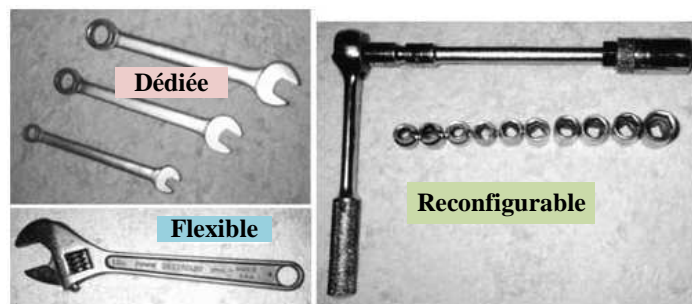


Figure 1.1.6. Exemple des outils dédiés, flexibles et reconfigurables.

En ce qui concerne les systèmes de production, le Tableau 1.1.2 présente les avantages et les limitations/inconvénients de chaque type de système de production DML, FMS et RMS (Koren *et al.*, 1999), (ElMaraghy, 2006), (Malhotra *et al.*, 2009), (Malhotra *et al.*, 2010).

	Avantages	Limitations ou Inconvénients
DML	<input checked="" type="checkbox"/> Débit élevé. <input checked="" type="checkbox"/> Coût/produit faible.	<input checked="" type="checkbox"/> Pas de flexibilité. <input checked="" type="checkbox"/> Capacité fixe. <input checked="" type="checkbox"/> Fonctionnalités fixes. <input checked="" type="checkbox"/> Pour un type unique de produits.
FMS	<input checked="" type="checkbox"/> Flexible. <input checked="" type="checkbox"/> Convertible. <input checked="" type="checkbox"/> Conçu pour une variété de produits.	<input checked="" type="checkbox"/> Onéreux. <input checked="" type="checkbox"/> Débit faible. <input checked="" type="checkbox"/> Coût/produit élevé.
RMS	<input checked="" type="checkbox"/> Réactif. <input checked="" type="checkbox"/> Intégrable. <input checked="" type="checkbox"/> Personnalisable. <input checked="" type="checkbox"/> Capacité évolutive. <input checked="" type="checkbox"/> Convertible.	<input checked="" type="checkbox"/> Difficulté de sélection des modules de base. <input checked="" type="checkbox"/> Mesure difficile de la reconfigurabilité.

Tableau 1.1.2. Avantages et limitations/inconvénients des DML, FMS et RMS.

Le DML est adapté lorsque la demande du produit dépasse la capacité de production. Quand la demande n'est pas stable, ce défi est théoriquement traité par le FMS qui a des capacités évolutives. Néanmoins, le FMS est une solution très coûteuse parce qu'il doit être capable de produire n'importe quel produit, et parfois avec des combinaisons différentes de produits, et dans n'importe quel ordre. En pratique, ce concept nécessite plus de fonctions que nécessaire, c'est donc un choix onéreux qui est une des principales raisons pour laquelle les FMS sont peu présents dans les industries.

Dans le contexte actuel, nous avons déjà évoqué le fait que les systèmes de production traditionnels DML et FMS ne peuvent pas répondre aux exigences dictées par le nouvel environnement concurrentiel mondial. La notion de système de production de type RMS réunit les caractéristiques positives des DML et des FMS, et en plus il améliore les points indésirables comme il est montré dans le Tableau 1.1.3 (Koren *et al.*, 1999), (Dashchenko, 2006) et (ElMaraghy, 2006) :

	DML	RMS	FMS
Structure du Système	Fixe	Modifiable	Modifiable
Structure de la Machine	Fixe	Modifiable	Fixe
Réglage du Système	Partie	Famille de composants	Machine
Flexibilité	Non	Personnalisée	Générale

Évolutivité	Non	Oui	Oui
Exploitation des outils simultanément	Oui	Oui	Non
Productivité	Elevée	Elevée	Faible
Coût	Faible	Moyen	Onéreux

Tableau 1.1.3. Comparaison entre les DML, FMS et RMS.

Enfin, il est possible de comparer les trois types de système de production DML, FMS et RMS en se basant principalement sur les trois éléments qui sont la capacité, la fonctionnalité et le coût.

Bien que DML et FMS soient limités en capacité/fonctionnalité, pour les RMS celles-ci sont changeantes au cours du temps en fonction du marché. Ainsi, un RMS oscille entre une DML et un FMS en termes de capacité et de fonctionnalité. Il a aussi la particularité d'avoir des capacités et des fonctionnalités variables. Effectivement, un RMS est celui dont la capacité de production est réglable aux fluctuations de la demande de produit, et dont la fonctionnalité est adaptable à de nouveaux produits (Mehrabi *et al.*, 2002a), (Dashchenko, 2006), (Koren, 2010a). En d'autres termes, les DML et FMS sont des systèmes statiques et le RMS est un système dynamique qui peut changer sa capacité et ses fonctionnalités en réponse aux changements du marché (Figure 1.1.7).

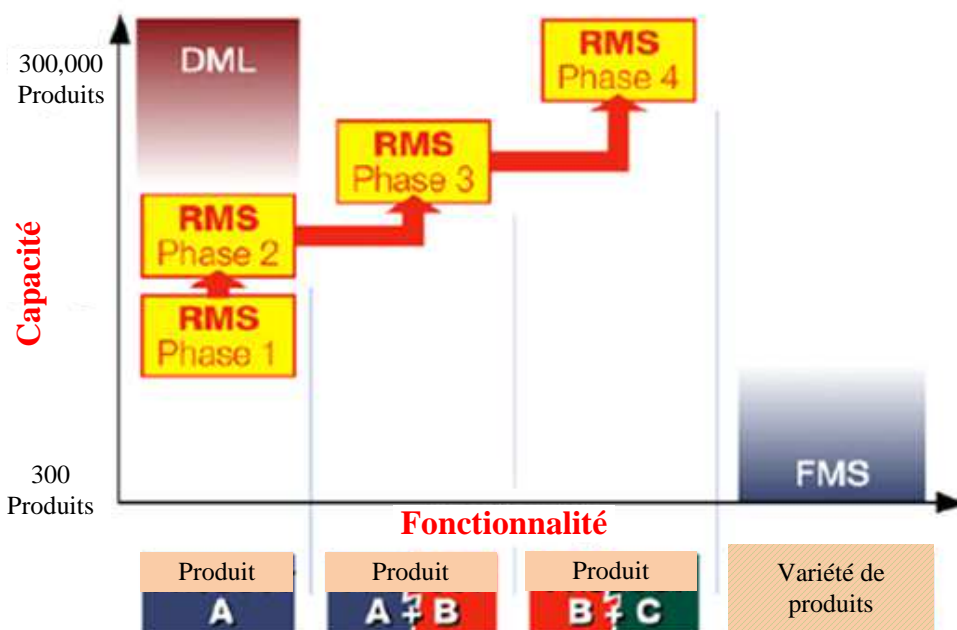


Figure 1.1.7. Capacité/Fonctionnalité pour les DML, FMS et RMS.

Enfin, en ce qui concerne les coûts par rapport à la capacité du système, les DML sont limités par leur capacité maximale fixe. Une ligne supplémentaire doit être construite quand la capacité actuelle ne suffit plus à répondre au marché. Par contre, le FMS est évolutif mais à un rythme constant en ajoutant des machines en parallèle. Enfin, les RMS sont complètement évolutifs à rythme variable qui dépend de la conception initiale du RMS et de la situation du marché (Figure 1.1.8). (Koren *et al.*, 1999), (Koren, 2010a).

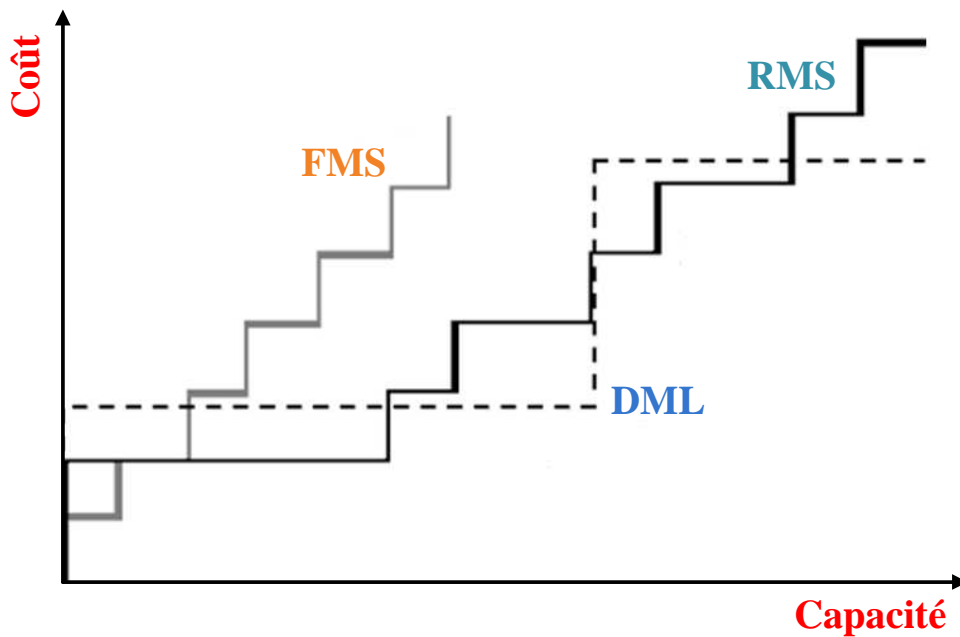


Figure 1.1.8. Capacité/Coût pour les DML, FMS et RMS.

Nous constatons que la flexibilité personnalisée d'un RMS est moins coûteuse que la flexibilité générale d'un FMS, les RMS sont donc des systèmes hautement productifs et rentables. Effectivement, la flexibilité personnalisée est la clé de l'intérêt du RMS par rapport au FMS.

En conséquence, si les DML permettent de produire toujours au moins 75% de sa capacité maximale, elle est la solution la plus économique en considérant que les changements de produits ne sont pas pris en compte. En outre, si seules de petites quantités sont nécessaires, alors les FMS sont les plus économiques. Par contre, si de grandes quantités sont nécessaires et couplés à une grande incertitude du marché, le RMS est la solution la plus rentable (Dashchenko, 2006).

Grâce à ces comparaisons détaillées, nous pouvons former un tableau récapitulatif (Tableau 1.1.4) contenant les trois systèmes de production permettant d'identifier précisément leurs caractéristiques et objectifs (Mehrabi *et al.*, 2000a), (ElMaraghy, 2006), (Malhotra *et al.*, 2009).

	Définition	Objectif
Lignes de Production Dédiées (DML)	C'est un système de production équipé d'une automatisation fixe et destiné à la production d'un type unique de produits avec un débit élevé.	L'objectif économique est de travailler à pleine capacité afin que le coût par produit soit relativement faible.
Systèmes de Production Flexibles (FMS)	Il s'agit d'un système de production qui est capable de produire une variété de produits avec un volume variable.	Le but économique d'un FMS est de permettre une meilleure rentabilité concernant une simple modification des conceptions actuelles de produits, pour répondre aux nouvelles exigences du marché sans ajouter d'investissements considérables dans le système de production actuel.
Systèmes de Production Reconfigurables (RMS)	Le RMS est un nouveau système de production créé de modules basiques (matériels et logiciels) qui peuvent être réorganisés et/ou remplacés afin d'ajuster la capacité et la fonctionnalité de la production, en répondant aux changements brusques du marché et/ou aux exigences régulières.	L'objectif d'un RMS peut être résumé par la déclaration : « Exactement la capacité et la fonctionnalité nécessaires, uniquement en cas de besoin ».

Tableau 1.1.4. Tableau récapitulatif des trois systèmes de production DML, FMS et RMS.

Le prochain paragraphe va permettre d'introduire la notion d'agilité dont l'objectif est de faciliter la reconfiguration et l'évolutivité du système.

6. Aspects de l'agilité

L'agilité est une propriété d'une personne, d'un objet, d'un système ou encore d'une entreprise. Ils peuvent être agiles ou non, ou ils peuvent avoir un certain degré d'agilité. Selon la littérature, on trouve différentes définitions de l'agilité :

- ✓ « Habilité à bouger rapidement de façon légère ».
- ✓ « Synonyme de légèreté de souplesse ».

- ✓ « Capacité à réagir promptement ».

Ces approches sont relativement générales et ont des implications dans de nombreux domaines tels que l'industrie, les transports, la logistique,...

Au niveau de l'industrie, l'agilité est une caractéristique qui a été identifiée comme un nouveau besoin généralement absent dans les entreprises ou les systèmes de soutien d'entreprise. Systématiquement, l'agilité est la capacité d'un système à se déployer dans un environnement évolutif et imprévisible en apportant une réponse efficace aux différents défis et exigences au sein d'une mission (Nagel, 1992).

Historiquement, la notion d'agilité de l'entreprise a vu le jour avec l'évolution rapide de la technologie et des marchés, les exigences imprévisibles du cycle de vie des équipements, et parce que les systèmes traditionnels sont incompatibles avec l'évolution continue des besoins. Ainsi, de nombreuses recherches ont été réalisées pour examiner sur 100 systèmes de types différents afin d'extraire les systèmes qui répondent efficacement à la notion d'agilité (Boss *et al.*, 2010) , (Dove, 2013a).

6.1. Métriques d'une réponse efficace

Dans la littérature, une réponse efficace est représentée par les quatre métriques suivantes (Dove, 2001), (Dove, 2013b) :

- ✚ Temps : la réponse doit être dans un temps convenable et suffisamment rapide pour offrir de la valeur ;

- ✚ Coût : le coût de cette réponse doit être abordable ;

- ✚ Préviation : la réponse efficace doit être prévisible, nous pouvons donc compter sur lui pour répondre aux attentes ;

- ✚ Globalité : cette réponse doit être compréhensive et complète pour couvrir tous les points importants au sein de la mission.

Le concept d'agilité permettra ainsi de réduire le temps nécessaire pour alimenter le marché avec des produits/services adéquates. De plus, la gestion des risques dans un environnement dominé par le changement et l'incertitude est une proposition de valeur pour les systèmes agiles.

En d'autres termes, nous pouvons définir l'agilité comme une réponse innovante à des situations imprévisibles, une notion nécessaire pour une gestion proactive des risques, et une extension du cycle de vie des équipements des systèmes.

6.2. Structure agile

Les experts des systèmes agiles, (Boss *et al.*, 2010) et (Dove, 2013b), évoquent le fait que l'intervention agile peut être réalisée par une structure agile qui a vu le jour d'après des recherches approfondies, et apparaît aussi bien nécessaire et suffisante.

Elle est reconnue comme une structure Glisser-Déposer (Drag&Drop), Brancher-Démarrer (Plug&Play) avec les trois éléments essentiels suivants (Figure 1.1.9) :

- **Modules encapsulés** : il s'agit d'un catalogue des composants réutilisables Drag&Drop avec l'aspect dans lequel ils appartiennent. Ces modules peuvent également contenir d'autres modules unitaires avec des interfaces conformes à l'infrastructure passive Plug&Play. Ainsi, ils peuvent être connectés module à module ou glissés et déposés dans un environnement qui contient d'autres modules connectés à travers l'infrastructure passive. De nouveaux modules peuvent être ajoutés à un groupe de modules, et aussi de nouveaux groupes de modules peuvent être ajoutés de manière asynchrone. Ces groupes de modules fournissent ainsi une variation et une diversité entre les modules afin de permettre l'augmentation de la capacité fonctionnelle d'un module principale comme, par exemple, le module de déploiement d'une configuration pour les systèmes de production reconfigurables.

- **Infrastructure passive** : cet élément assure la connectivité entre les modules en respectant des règles et normes qui permettent et contraignent l'exploitation Plug&Play. Par exemple, les règles d'utilisation, d'ajout et de combinaison des modules. Généralement, l'infrastructure passive évolue selon les nouveaux besoins.

- **Infrastructure active** : un système agile n'est pas une chose conçue et déployée dans un environnement fixe, puis laissée seule. L'agilité est d'être réactive avec les nouvelles exigences. L'infrastructure active doit donc être désignée d'une façon qui soutient le fonctionnement agile, et elle peut être traduite par les modules responsables suivants :

- **Module de combinaison** : ce module est responsable d'assurer que les nouveaux composants créés/ajoutés sont mis à jour dans un délai suffisant pour répondre aux besoins de réponse ;

- **Module de préparation** : les modules disponibles doivent toujours être en état déployables. Effectivement, ce module est chargé d'assurer que les modules nécessaires sont prêts pour le déploiement ;

- **Module d'assemblage** : ce module permet d'assembler les nouvelles configurations du système lorsque celui-ci doit être reconfiguré ;

- **Module d'évolution** : la collection de modules disponibles doit être régulièrement évoluée pour s'adapter à ce qui est nécessaire. Egalement, l'infrastructure passive doit être évolutive afin de prendre en compte les nouvelles normes et règles exigées par des nouvelles configurations.

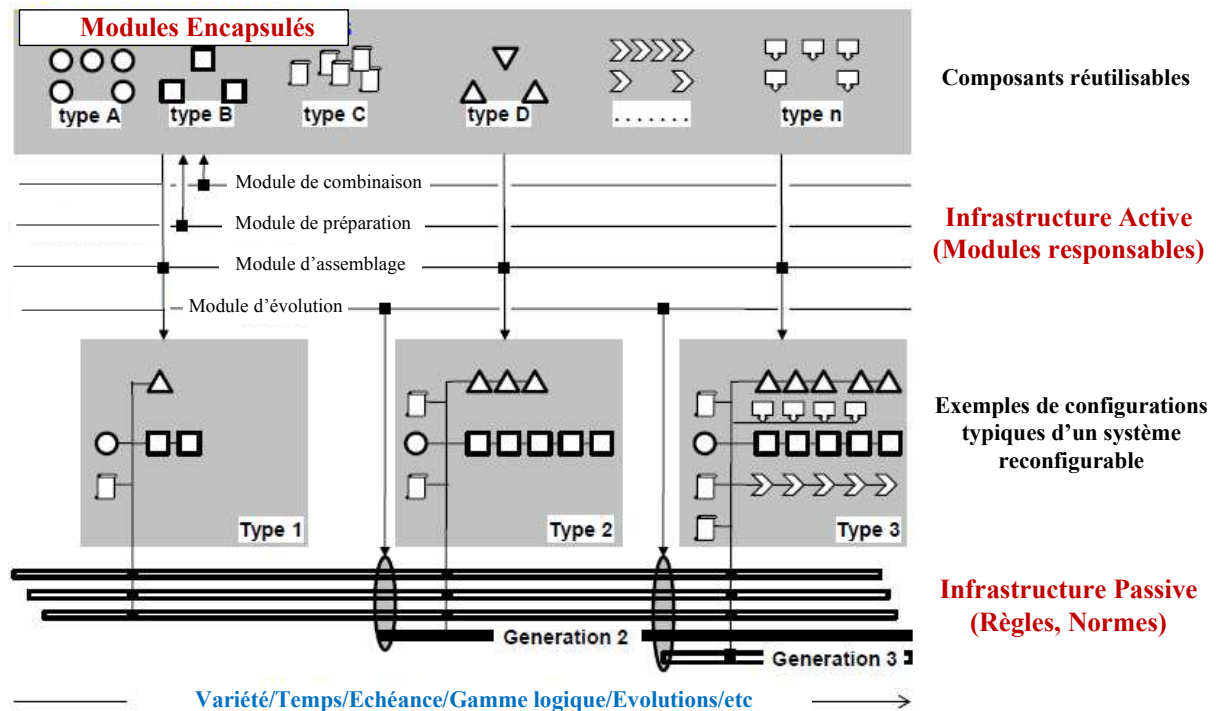


Figure 1.1.9. Modèle d'une structure agile.

6.3. Principes fondamentaux de l'agilité

Un système agile est construit pour permettre et faciliter la reconfiguration et l'évolutivité du système afin de faire face à des situations imprévisibles. Les principes de conception réutilisable, de reconfiguration et d'évolution s'intègrent avec la structure agile, et sont brièvement détaillés ci-dessous (Boss *et al.*, 2010), (Dove, 2013b) :

Principes de conception réutilisable :

- Modules encapsulés: les modules sont distincts, séparables, faiblement couplés, ce sont des unités autosuffisantes qui coopèrent dans un but commun ;
- Facilité d'interfaçage: les modules sont définis selon les normes d'interface, et sont facilement insérés ou retirés ;
- Facilité de réutilisation : les modules sont facilement réutilisables et reproductibles.

Principes de reconfiguration :

- Interaction entre pairs : les modules communiquent directement par pairs, les relations parallèles sont plutôt favorisées par rapport aux relations séquentielles ;
- Information et commande distribuées: l'information est associée localement, accessible globalement, et diffusée librement. En outre, les décisions sont prises au point de savoir maximum ;

■ Auto-organisation: les relations entre modules sont autodéterminées lorsque c'est possible, et l'interaction du module est auto-réglable.

Principes d'évolution :

- Normes évolutives: l'infrastructure passive normalise les communications et les interactions entre les modules et définit la compatibilité du module ;
- Redondance et diversité: le double emploi des modules fournit une capacité large d'options, et la diversité parmi les modules similaires qui emploient des méthodes différentes est exploitée ;
- Capacité de flexibilité: les modules peuvent être combinés dans des assemblages réagissant à l'augmentation et à la diminution de la capacité fonctionnelle au sein de la structure actuelle.

En plus des exigences du système fonctionnel, l'analyse situationnelle de l'intervention permet d'identifier les besoins de réaction qui sont transmis à la conception de la structure indiquant le caractère nécessaire des modules et des groupes de modules qui aident à leur tour à identifier la nature nécessaire de l'infrastructure passive et active. Ces besoins peuvent ne pas être directement présents dans les exigences du client, mais ils sont nécessaires pour la conception d'une structure efficace. Contrairement aux exigences fonctionnelles saisies normalement dans des déclarations spécifiques, les exigences d'intervention ont besoin d'une diversité suffisante pour aboutir à une capacité qui peut répondre à des nouvelles situations. Le cadre efficace pour structurer l'analyse situationnelle d'une réponse entraîne une réflexion analytique en domaines réactifs et proactifs (**Dove, 2001**), (**Sillitto, 2013**).

6.4. Systèmes agiles et Lean

Dans la littérature, nous constatons des confusions sur les significations des systèmes agiles et Lean. Ceux qui ont des tendances Lean pensent trop souvent que les concepts agiles sont inclus dans la notion générale de Lean. De même, ceux qui sont dans le groupe agile pensent parfois l'inverse. Effectivement, ces deux mots ont été choisis de façon significative pour désigner deux types différents d'intérêts fondamentaux, ces deux concepts sont basés sur des objectifs de processus de valeur tels que le système Lean se concentre sur le fonctionnement efficace du système dans une configuration donnée alors que le système agile met l'accent sur la transformation efficace du système en créant une capacité facultative de réaction aux nouvelles exigences (**Dove, 2013a**).

Ces deux systèmes sont différents, mais il y a certains points communs issus des meilleures pratiques de chacun. Le système Lean évalue l'efficacité de fonctionnement et fonctionne principalement à partir de principes opérationnels (le temps désignant le rythme de production, le flux des pièces, le changement rapide d'outils, l'intégration de la logistique,...). Le système agile évalue la capacité de

réaction efficace aux différents changements et fonctionne principalement à partir de principes structuraux (Figure 1.1.10).

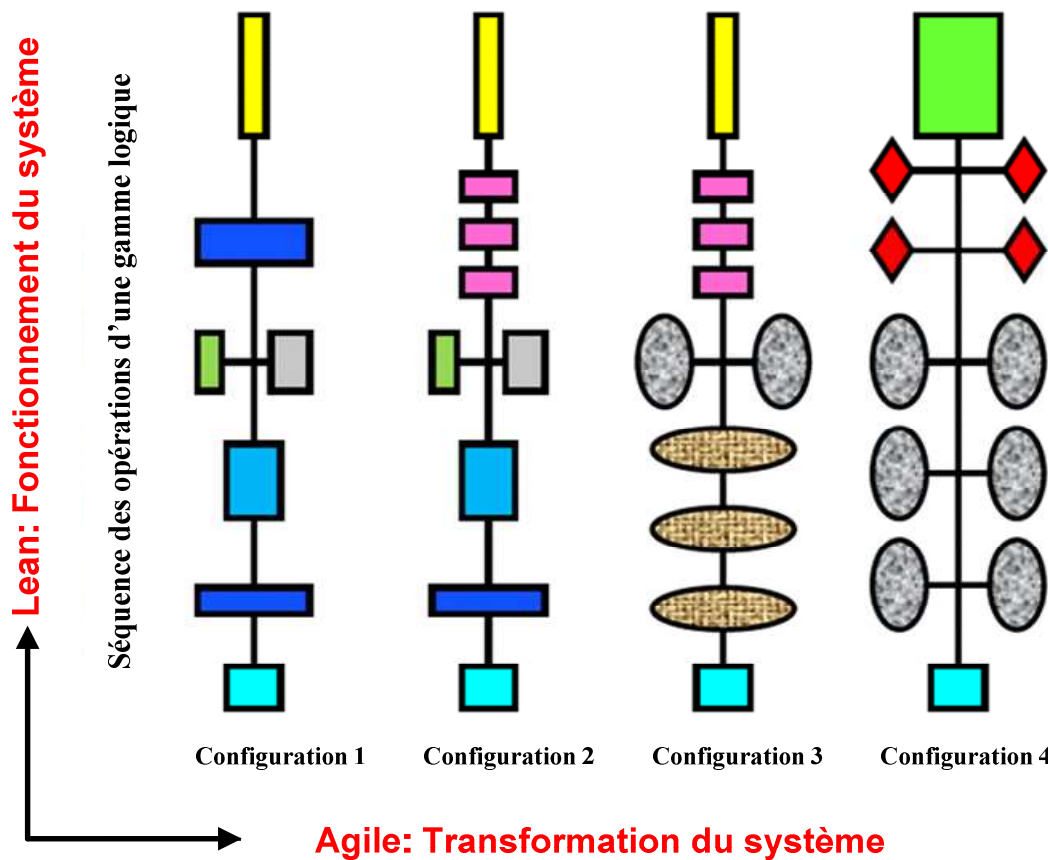


Figure 1.1.10. Systèmes agiles et Lean.

7. Conclusion

L'objectif de ce chapitre était de présenter un état de l'art concernant l'évolution des systèmes de production des lignes fixes et dédiées vers l'aspect de l'agilité. Après avoir défini généralement le système de production et les différents types de solutions pour l'organisation du système, des comparaisons ont été réalisées au travers de plusieurs critères. Nous avons proposé aussi différentes définitions de l'agilité.

De plus, nous avons constaté que le RMS est la clé du futur pour l'industrie parce qu'il est un système dynamique qui peut changer sa capacité et ses fonctionnalités en réponse aux changements du marché. De plus, le concept d'agilité nous permet de réduire le temps nécessaire pour reconfigurer le système et atteindre le marché avec des produits/services adéquates. L'agilité est également une réponse innovante à des situations imprévisibles. Dans le future, un système de production reconfigurable et agile sera un concept innovant qui permettra de combiner la reconfigurabilité et

l'agilité d'un système de production en réagissant efficacement aux exigences imposées du marché, des clients, de la technologie des procédés, de l'environnement et de la société afin que l'entreprise soit dynamique, compétitive et rentable.

Dans le chapitre suivant, nous présentons une classification détaillée des configurations d'un RMS ainsi qu'une méthode mathématique pour calculer le nombre de configurations nécessaire pour répondre à la demande du marché.

Chapitre 2

Classification des configurations d'un système de production

La notion de reconfiguration a vu le jour au début des années 1990, elle est utilisée de manière variée dans des domaines aussi différents que la mécanique, l'électronique, l'informatique, l'automatique, les communications et les systèmes de production.

L'objectif de ce chapitre est de présenter une classification détaillée des différentes configurations possibles d'un système de production en nous concentrant sur les cas de configurations de type RMS.

Sommaire

1. Introduction	61
2. Calcul du nombre total de configurations	61
3. Configurations symétriques/asymétriques	63
4. Calcul du nombre de configurations de type RMS	66
5. Conclusion	67

1. Introduction

La reconfiguration nous entoure dans notre vie quotidienne dans nos maisons, nos chambres et nos bureaux dont nous modifions souvent l'utilisation des pièces afin d'adapter l'usage de chacun à des nouveaux besoins ou à des nouvelles envies. De même, au niveau de l'informatique personnelle, la reconfiguration d'un ordinateur individuel peut concerner à la fois le matériel et/ou le logiciel qui doivent être mis à jour. De la même manière, pour n'importe quel système de n'importe quel domaine, sa structure physique ainsi que son organisation logique peuvent être entièrement modifiées afin de s'adapter aux nouvelles contraintes imposées par les missions demandées et l'environnement. La reconfiguration apparaît donc comme un moyen permettant de satisfaire au mieux les besoins et les contraintes imposées au système.

Le chapitre précédent a permis de décrire en détail l'évolution des systèmes de production des lignes dédiées vers les systèmes agiles. Différentes comparaisons entre tous les systèmes de production y ont également été présentées.

Ce chapitre a donc pour objectif de présenter une classification des différentes configurations possibles d'un système de production en nous focalisant sur les configurations symétriques les plus générales et surtout les configurations de type RMS. Nous proposons également une méthode mathématique pratique qui permet de calculer le nombre total de configurations possibles et plus précisément le nombre des configurations de type RMS basées sur le nombre de machines disposées en plusieurs phases.

2. Calcul du nombre total de configurations

Tout d'abord, la classification des configurations nécessite la détermination du nombre de configurations possibles pour un cas donné.

Lorsque la demande quotidienne Q (produit/jour) et le temps total de toutes les opérations pour chaque produit t (min/produit) sont donnés. Nous pouvons calculer le nombre minimum de machines N nécessaire dans le système pour répondre à la demande, ce nombre N est alors calculé par l'équation suivante (qui doit être arrondi à l'entier supérieur) (**Koren et al., 2010b**) :

$$N = \frac{Q \times t}{t_j \times f} \quad \text{Equation (1)}$$

Où : f représente la fiabilité de la machine (on peut généralement supposer que la fiabilité est égale à 100% pour toutes les parties des équipements, c.à.d. $f = 1$).

t_j : le temps de travail par jour (min/jour).

Par exemple, prenant le cas d'une entreprise qui a besoin de produire 600 produits par jour et sachant que le temps de travail nécessaire pour chaque produit

est de 7.5 min. Supposons de même que le temps de travail de l'entreprise soit de 960 min/jour, la fiabilité est de 100% pour toutes les parties des équipements, et l'ensemble de machines est capable de faire toutes les opérations. Alors, cinq machines au moins sont nécessaires dans ce cas.

En général, le nombre total de configurations augment exponentiellement par rapport au nombre de machines (Figure 1.2.1) selon les experts (**Zhu, 2005**) et (**Koren et al., 2010b**).

La question la plus importante est quelle est alors la meilleure façon d'organiser et de connecter ces machines pour rendre la production optimale? Par exemple, devraient-elles être disposées dans une ligne en série ou sous la forme d'un système purement parallèle, ou encore une combinaison de ces deux dernières? Laquelle de toutes les configurations possibles est la plus avantageuse? Dans le cas précédent de cinq machines, le nombre total de configurations est 48 selon la courbe logarithmique ci-dessous, et quelques configurations sont illustrées sur la Figure 1.2.2 (**Koren et al., 2010b**). Donc, une classification basée sur l'arrangement et les connexions entre les machines est nécessaire et utile afin de mettre en œuvre une méthode efficace pour arranger et relier les machines. En d'autres termes, l'arrangement et les connexions entre les machines seront des critères principaux pour évaluer les configurations (**Shpitalni et al., 2004**).

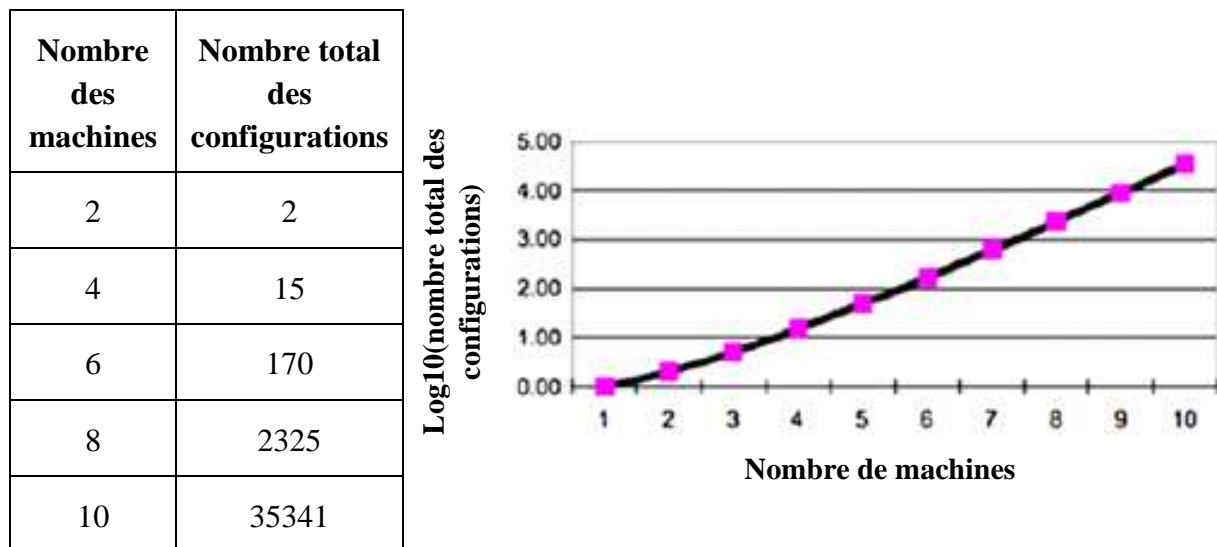


Figure 1.2.1. Nombre total de configurations en fonction du nombre des machines.

Dans la suite de ce chapitre, nous donnons une classification détaillée des différentes configurations possibles d'un système de production en nous concentrant sur les cas de configurations de type RMS.

3. Configurations symétriques/asymétriques

Par définition, une configuration d'un système de production est une organisation temporaire qui prend fin soit parce que les objectifs fixés sont atteints, soit parce que les contraintes appliquées au système ne sont plus respectées. Une bonne organisation doit respecter les délais de traitements entre les postes/machines au sein de chaque phase et entre les phases elles mêmes.

Dans la littérature, (Spicer *et al.*, 2002), (Koren, 2010a) et (Koren *et al.*, 2010b), les configurations sont classées soit comme *symétrique* ou comme *asymétrique*. Cette classification est généralement basée sur l'existence ou non d'un axe de symétrie au long de la structure de la configuration (Figure 1.2.2).

Malheureusement, les configurations asymétriques ajoutent une grande complexité et le nombre de configurations asymétriques est beaucoup plus grand que le nombre de configurations symétriques. Pour les 5 machines dans l'exemple précédent, nous avons 48 configurations dont 30 configurations asymétriques. Il est important de noter que les configurations *d'* et *e'* (Figure 1.2.2) sont classées comme asymétriques même si elles ont un axe de symétrie car elles peuvent être positionnées différemment comme *d* et *e*. De même, les deux configurations *f* et *g* sont classées comme des configurations asymétriques alors qu'elles peuvent être aussi classées dans la catégorie symétrique. Ainsi, les configurations asymétriques ne sont généralement pas utilisées.

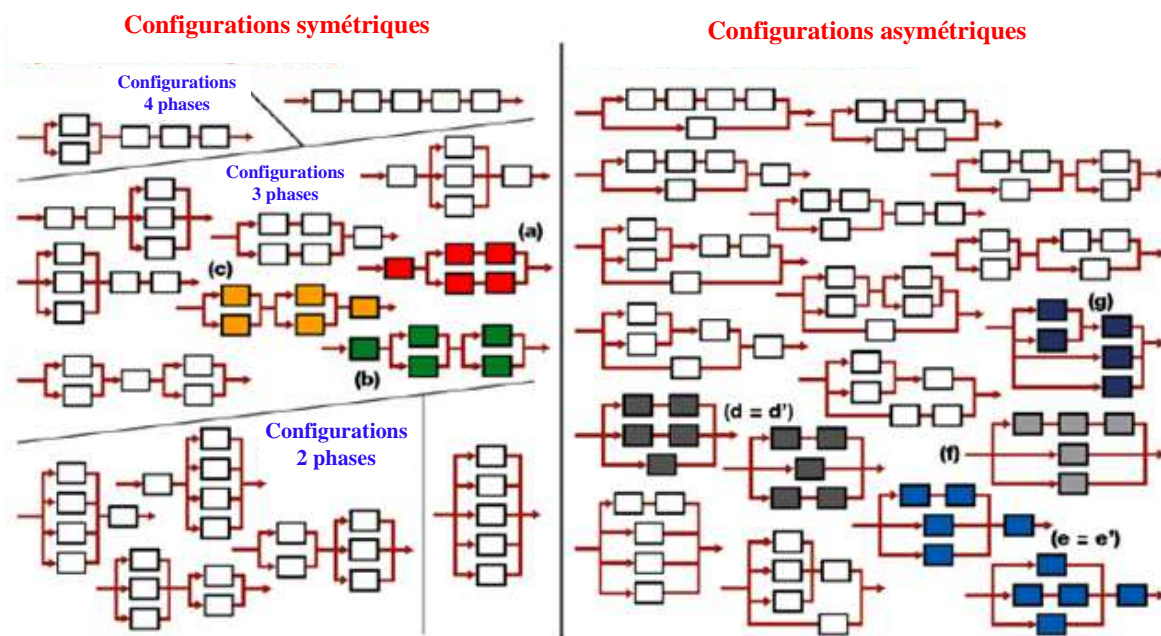


Figure 1.2.2. Quelques configurations pour 5 machines.

Par ailleurs, les configurations asymétriques peuvent être sous classées en **configurations avec processus variable** ou en **configurations avec un seul processus** où les machines sont non identiques au moins par une des phases. Des exemples correspondants sont présentés sur la Figure 1.2.3.

Les configurations avec un processus variable sont caractérisées par des chemins d'écoulement non-identiques. Elles ont donc besoin de plusieurs schémas de processus avec leurs démarches correspondantes. Par exemple, le système représenté dans le cas A, a un certain nombre de chemins d'écoulement possibles *a-b-c-d-e*, *g-c-f*, *g-c-d-e*, etc. En réalité, ces configurations ne sont pas pratiques parce que les concepteurs ne sont pas habitués aux schémas de processus multiples pour la fabrication d'un même produit. De même les différents schémas de processus et les chemins d'écoulement correspondants augmentent les problèmes de qualité et rendent la détection de l'erreur de qualité plus compliquée.

Bien que la planification de processus soit identique dans chaque chemin d'écoulement dans la deuxième classe de configurations asymétriques (cas B), les machines sont différentes au moins dans une phase. Par exemple, la machine *d* de la phase 4 doit être deux fois plus rapide que la machine *c*.

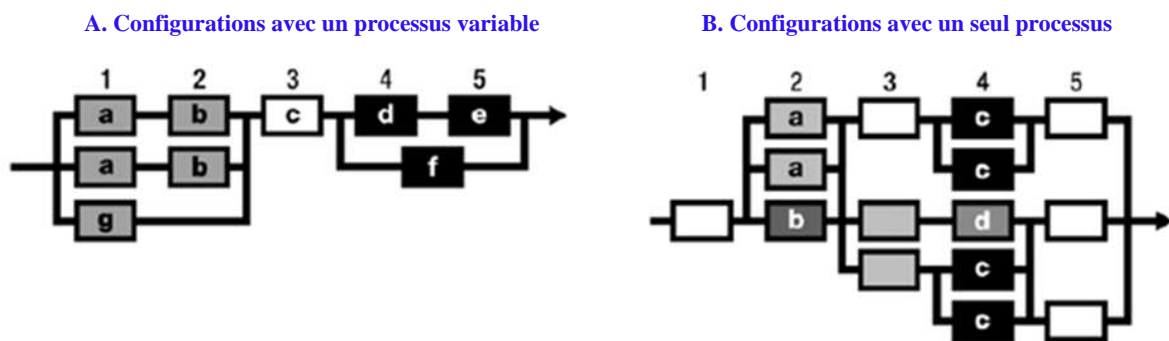


Figure 1.2.3. Deux classes des configurations asymétriques.

Par contre, en ce qui concerne les configurations symétriques, les machines qui constituent chacune des phases sont identiques et les délais de traitement de chaque machine dans une phase particulière sont donc égaux. D'un autre côté, si l'on prend les deux configurations symétriques *a* et *b* (Figure 1.2.2) qui ont des arrangements identiques pour les postes (un poste dans la phase 1, deux postes dans la phase 2, et deux postes dans la phase 3) celles-ci diffèrent en raison des différentes connexions entre phases (la configuration *b* utilise un couplage croisé entre les phases 2 et 3).

De même, les configurations symétriques peuvent être classées en trois classes de base, comme indiqué dans la Figure 1.2.4 (Koren *et al.*, 2010b).

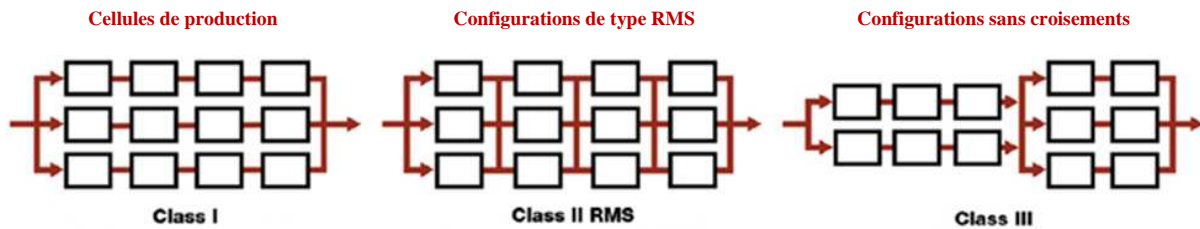


Figure 1.2.4. Trois classes des configurations symétriques.

Dans un contexte de production réel, il est alors recommandé que ces trois classes de configurations symétriques soient considérées, il s'agira alors toujours de configurations avec un seul processus et avec des machines identiques dans chaque phase :

✓ **Classe I : Cellules de production**

Il s'agit de configurations comprenant plusieurs lignes de production en série disposées en parallèle et sans croisements. Cette classe de configurations est généralement simple et elle est beaucoup utilisée au Japon.

✓ **Classe II : Configurations de type RMS**

Ceux sont des configurations avec des connexions croisées après chaque phase. Une pièce de n'importe quelle machine à la phase i peut être transférée à n'importe quelle machine à la phase $(i + 1)$. Toutes les machines et les opérations de chaque phase sont identiques. Les trois producteurs automobiles américains internationaux utilisent ces configurations dans l'usinage de leurs composants de motopropulseur. Il s'agit d'un système typique composé de 15 phases avec 6 machines par phase.

✓ **Classe III : Configurations sans croisements**

Cette catégorie comprend les combinaisons des deux classes précédentes qui possèdent des phases sans croisements.

Par conséquent, la configuration la plus générique est la configuration de type RMS, la Figure 1.2.5 représente un système RMS réel composé de 3 phases avec des supports mobiles (robots, convoyeurs, ...) qui transportent les pièces entre les machines de différentes phases. Généralement, un support mobile principal transfère une pièce vers un support mobile secondaire qui peut la prendre pour le traitement dans une des machines de la phase. Une fois que la pièce a été traitée, ce support mobile secondaire renvoie la pièce au support principal qui, à son tour, la transfère de nouveau vers une nouvelle phase où le prochain support mobile secondaire peut prendre la pièce pour effectuer le traitement nécessaire, et ainsi de suite.

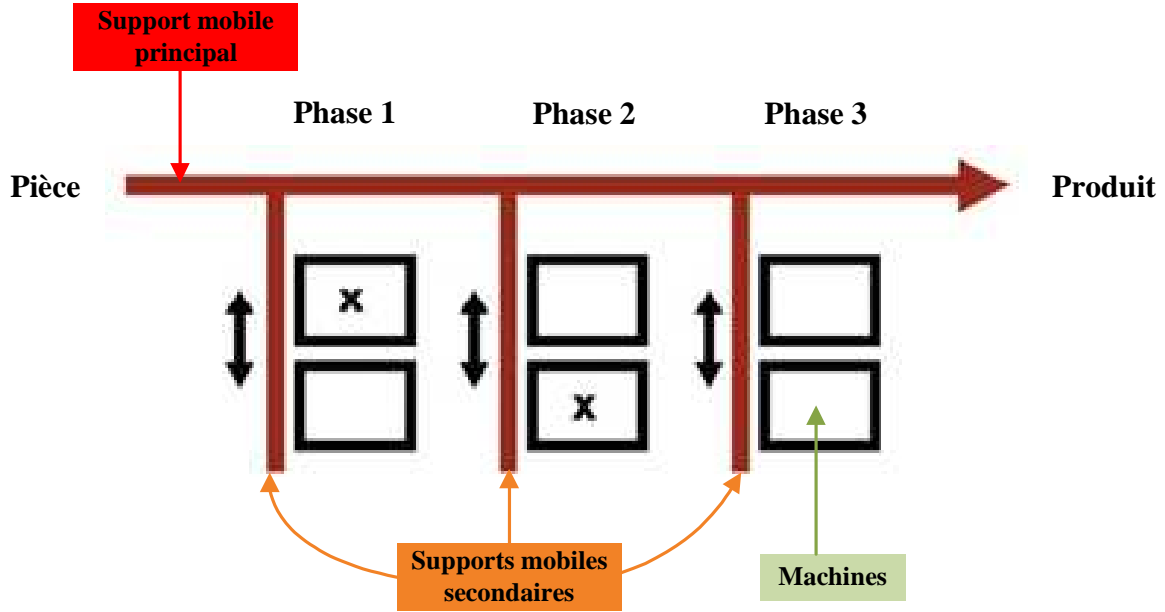


Figure 1.2.5. Système RMS pratique.

4. Calcul du nombre de configurations de type RMS

Dans ce chapitre, nous avons vu que le nombre minimum de machines N requis dans le système peut facilement être calculé en résolvant l'équation 1. De plus, comme déjà indiqué précédemment, le nombre de toutes les configurations possibles avec N machines est trop grand.

Dans la littérature, et après une étude mathématique approfondie des reconfigurations d'un système, les auteurs experts de RMS (**Koren et al., 2010b**) concluent sur le point suivant :

« Les équations pour calculer le nombre de configurations avec N machines n'existent que pour les configurations de type RMS (Classe II) ».

Ces équations de base sont données sous deux formes différentes :

- 1- Le nombre de configurations de type RMS avec N machines disposées jusqu'à $m=N$ phases est donné par :

$$k = \sum_{m=1}^N \binom{N-1}{m-1} = 2^{N-1} \quad \text{Equation (2)}$$

- 2- Le nombre de configurations de type RMS avec N machines disposées exactement en m phases est donné par :

$$k = \binom{N-1}{m-1} = \frac{(N-1)!}{(N-m)!(m-1)!} \quad \text{Equation (3)}$$

En outre, les experts proposent une méthode mathématique pratique afin que les ingénieurs puissent facilement développer des configurations recommandées pour les systèmes de production reconfigurables RMS. Les résultats mathématiques de ces deux équations pour tout N et m peuvent être réorganisés dans un format triangulaire, connu sous le nom de triangle de Pascal.

En conséquence, nous constatons que le nombre de configurations de type RMS est donc beaucoup plus petit par rapport au nombre total de configurations possibles, comme indiqué dans le Tableau 1.2.1 suivant (Zhu, 2005) :

Nombre des machines (N)	Nombre total des configurations possibles	Nombre des configurations de type RMS (jusqu'à $m=N$ phases)
2	2	2
4	15	8
6	170	32
8	2325	128
10	35341	512

Tableau 1.2.1. Comparaison entre le nombre total des configurations possibles et le nombre de configurations de type RMS pour un nombre donné de machines.

Finalement, ce nombre devient encore plus petit lorsque le nombre de phases est donné, par exemple pour $N=6$ machines disposées exactement en $m=3$ phases, nous avons $k = 10$ configurations de type RMS.

Il est important de noter que dans un contexte industriel, de nombreuses contraintes suffisamment fortes réduisent aussi de manière importante le nombre de configurations possibles et réduisent ainsi très largement le champ des possibilités.

5. Conclusion

Ce chapitre a présenté une classification des différentes configurations possibles d'un système de production en nous focalisant sur les configurations de type RMS.

Une méthode mathématique pratique basée sur le nombre de machines disposées en plusieurs phases est utilisée dans le calcul du nombre de configurations de type RMS.

Le chapitre suivant présente les méthodes et les langages les plus récurrents de la modélisation d'un système qui permettent de faciliter l'étude et la communication entre les divers intervenants au sein d'un projet. Le langage de modélisation de systèmes complexes (SysML : System Modeling Language) est introduit, défini et spécifié.

Chapitre 3

Méthodes et langages de modélisation d'un système

Le recours à la modélisation est depuis longtemps une pratique indispensable afin d'analyser, comprendre et étudier un système ou encore un phénomène physique. Un modèle est une représentation abstraite d'un système complexe qui permet de le simplifier pour mieux le comprendre. Ainsi, dans ces travaux, nous avons besoin d'un outil de modélisation qui nous permet de spécifier ce nouveau type de systèmes de production nommé « Système de Production Reconfigurable et Agile (SPRA) ».

Par conséquent, ce chapitre rappelle les outils et les méthodes traditionnels de la modélisation d'un système. Nous caractérisons, ensuite, le langage de modélisation unifié (UML) en montrant son architecture, ses extensions et ses faiblesses en ingénierie système. Finalement, nous introduirons le langage de modélisation de systèmes complexes (SysML) qui est une réponse aux besoins de l'ingénieur système pour gérer la complexité croissante des systèmes.

Sommaire

1. Introduction	71
2. Outils et méthodes traditionnels de modélisation d'un système	71
3. Le langage UML.....	72
3.1. Historique	72
3.2. Caractéristiques	73
3.3. Architecture d'UML 2.0	74
3.4. Extensions et lacunes d'UML.....	76
4. Le langage SysML.....	77
4.1. Origines de SysML.....	78
4.2. Critères de SysML.....	79
4.3. Présentation synthétique de SysML	80
5. Conclusion	83

1. Introduction

L'Ingénierie Système (IS) est une démarche méthodologique générale qui englobe l'ensemble des activités adéquates pour concevoir, faire évoluer et vérifier un système apportant une solution économique et performante aux besoins d'un client tout en satisfaisant l'ensemble des parties prenantes. Depuis longtemps, les ingénieurs systèmes ont utilisé des techniques de modélisation **(Roques, 2009)**.

En outre, les systèmes devenant de plus en plus complexes, leur compréhension et leur maîtrise globale dépassent les capacités d'un seul individu. Il est souvent difficile de garder une vision globale et cohérente de la structure de l'ensemble et surtout des interactions internes au système.

Ainsi, la modélisation est une pratique indispensable qui permet de représenter de manière abstraite un système complexe pour le simplifier afin de faciliter sa compréhension. Le rôle de la modélisation est donc de représenter le système à différents niveaux d'abstraction pour bien appréhender, conceptualiser, concevoir, estimer, simuler, valider et justifier des choix au niveau du système, et également pour faciliter les communications entre les différents acteurs d'un projet. Par contre, il est important de noter que la modélisation décrit le système sans connaître ou maîtriser forcément toutes les lois réelles du système. Finalement, la modélisation décrit « quoi » et non « comment » **(OMG, 2009)**.

Par définition, la modélisation est une manière de décrire le système de sorte que certaines propriétés soient visibles afin d'analyser le système dans le but de le comprendre et de communiquer avec les membres d'une même équipe. En d'autres termes, la modélisation est un outil majeur de communication entre les divers intervenants au sein d'un projet. Chaque membre de l'équipe, depuis les utilisateurs jusqu'aux développeurs, utilisent et enrichissent le modèle différemment **(Roques, 2002), (Endres et al., 2003)**.

Dans ce chapitre, les outils et les méthodes traditionnels de la modélisation sont rappelés. Ensuite, nous caractériserons le langage de modélisation unifié (UML). Un autre objectif de ce chapitre sera d'introduire et de décrire le langage de modélisation le plus récent, le langage de modélisation de systèmes complexes (SysML), en présentant synthétiquement ses différents diagrammes.

2. Outils et méthodes traditionnels de modélisation d'un système

L'évolution d'outils de modélisation d'un système est détaillée et illustrée en **Annexe A**. Ainsi, nous pouvons citer quelques outils ou méthodes traditionnels les plus récurrents dans le milieu industriel et qui continuent d'être utilisés aujourd'hui : la méthode APTE, la méthode basée sur IDEF, la méthode MERISE, la méthode

GRAI, la méthode CIMOSA, les réseaux de Pétri, la méthode SADT, la méthode SA-RT, la méthode BOOCH, la méthode OMT et la méthode OOSE.

3. *Le langage UML*

Ces techniques traditionnelles sont limitées par leur portée et leur expressivité ainsi que par la difficulté de leur intégration avec d'autres formalismes, ainsi qu'avec les exigences système.

Les expériences ont montré que certaines faiblesses devaient être comblées pour en faire un langage commun utilisable par toutes les méthodes, adapté à toutes les phases du développement et compatible avec toutes les techniques de réalisation. L'intérêt d'un standard de modélisation universel est le passage à la modélisation haut niveau, au développement basé sur composants et à une notation unifiée pour toutes les méthodologies.

Le langage de modélisation unifié UML (Unified Modeling Language) se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, concevoir des solutions et communiquer des points de vue. Le langage UML, depuis sa création dans les années 1990, est devenu le langage de la modélisation pour les ingénieurs logiciels impliqués dans la création de systèmes logiciels orientés objets (OMG, 2013).

3.1. *Historique*

L'élaboration d'UML a commencé en 1994 par Grady Booch et Jim Rumbaugh qui ont entrepris d'unifier la méthode BOOCH et la méthode OMT. En 1995 la version 0.8 d'UML a été publiée. Puis en 1996, Ivar Jacobson a rejoint l'équipe et sa méthode OOSE a été ainsi fusionnée avec UML en donnant lieu à UML 0.9. Cette collaboration avait eu pour but de donner une certaine stabilité aux technologies orientée objet en fournissant un langage unifié et plus riche que l'existant, et qui pourra être utilisé par un large public (Booch *et al.*, 2007).

Ensuite, le groupe de gestion d'objets (OMG: Object Management Group) a rassemblé une équipe de travail composée d'industriels du logiciel pour élaborer et standardiser UML. La version UML 1.0 a ainsi vu le jour puis la version 1.1 a été adoptée en 1997. L'évolution d'UML a continué sous l'égide de l'OMG avec la version 1.3 qui a été publiée en 1999 et ensuite la version 1.4 qui a été adoptée en 2001.

Le langage UML était, tout au moins dans ses premières versions, peu adapté à la modélisation des systèmes complexes et donc au support de l'ingénierie système. La version 2 d'UML, officialisée en 2005, a introduit plusieurs niveaux de concepts et de diagrammes utiles pour l'IS. La Figure 1.3.1 représente cette élaboration d'UML. (Anvar *et al.*, 2006), (Piechocki, 2009), (OMG, 2013)

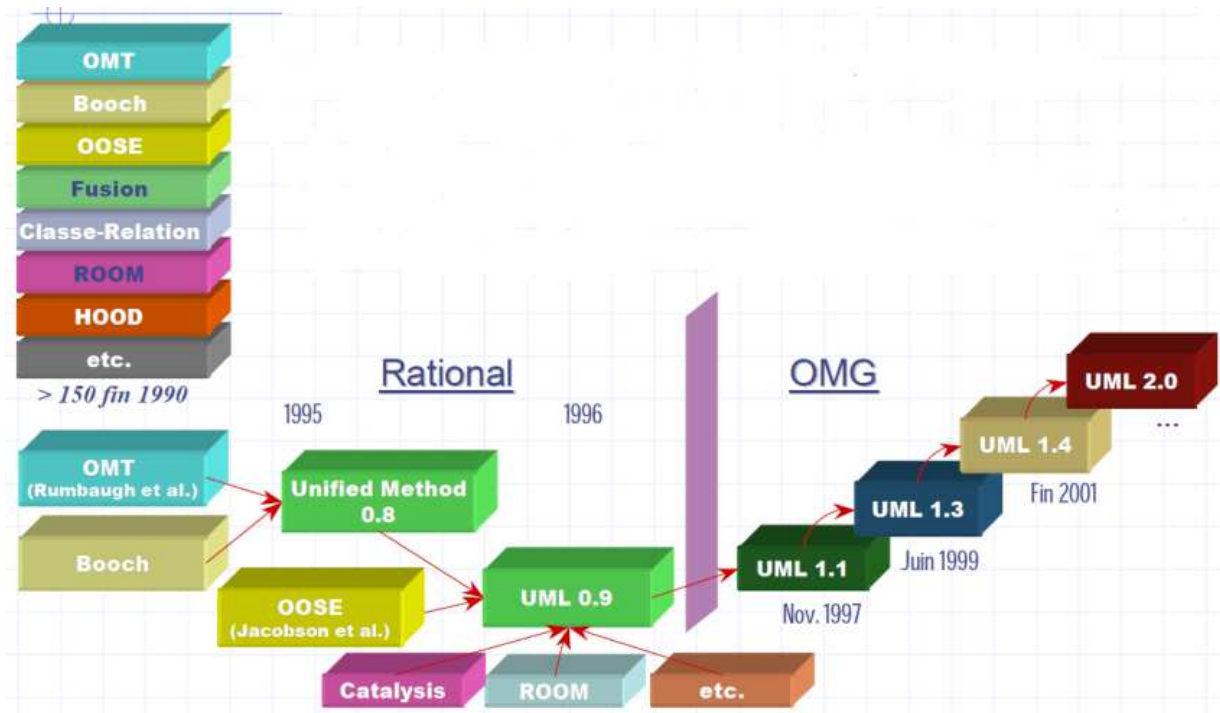


Figure 1.3.1. Elaboration d'UML.

3.2. Caractéristiques

Il est important de noter qu'UML est un langage et n'est pas directement associé à une méthode. De plus, UML peut être adaptable à tous les processus pouvant exciter l'intérêt suivant le domaine d'application et la nature du projet (**Booch et al., 2000**), (**Pilone et al., 2008**), (**Audibert, 2009**), (**Roques, 2011**).

La version 2 d'UML a été architecturée en prenant en considération les 5 principes globaux : (**UMLInfra, 2004**), (**OMG, 2013**)

✓ **La modularité** : les éléments du langage sont regroupées dans des packages dont la cohésion est maximisée et dont le couplage est minimisée.

✓ **Une architecture en couches** : elle est visible en quatre couches pour représenter les niveaux d'abstractions de langages. Cette architecture se caractérise par la hiérarchie des packages qui sépare les éléments de base des éléments de haut niveau qui les utilisent.

✓ **Le partitionnement** : il est utilisé pour organiser les différents domaines conceptuels au sein de la même couche. Dans ce cas, les packages de la superstructure d'UML sont partitionnés de manière modulaire, et les packages de l'infrastructure sont partitionnés de manière à faciliter la création de nouveaux langages à partir du méta-langage d'UML.

✓ **L'extensibilité** : la modularité et le partitionnement favorisent l'extension par création de nouveaux langages basés sur l'infrastructure d'UML. De plus, des mécanismes d'extension permettent l'adaptation d'UML à des domaines particuliers.

✓ **La réutilisation** : la modularité améliore la réutilisation des packages de l'infrastructure ainsi que de la superstructure.

3.3. Architecture d'UML 2.0

Généralement, il est très difficile de donner une représentation graphique complète d'un système. Par contre, il est possible de donner sur un tel système des vues partielles dont la conjonction donnera une idée utilisable en pratique sans risque d'erreur grave.

UML 2 s'articule autour de treize types de diagrammes, chacun d'eux étant dédié à la représentation des concepts particuliers d'un système logiciel. Ces types de diagrammes sont répartis par l'OMG en deux grands groupes, structurel et comportemental (Roques, 2009) :

Six diagrammes structurels :

1. Diagramme de classes : ce diagramme est généralement considéré comme le plus important de la modélisation orienté objet. Il représente l'architecture conceptuelle du système, il décrit les classes que le système utilise, ainsi que leurs liens. Enfin, il s'agit d'une vue statique car il ne tient pas compte du facteur temporel dans le comportement du système.

2. Diagramme d'objets : il montre les instances des éléments structurels et leurs liens à l'exécution. Ce diagramme permet d'éclairer un diagramme de classes en l'illustrant par des exemples. Il peut être utilisé pour vérifier l'adéquation d'un diagramme de classes à différents cas possibles.

3. Diagramme de packages : il montre l'organisation logique du modèle et les relations entre packages.

4. Diagramme de structure composite : il montre l'organisation interne d'un élément statique complexe.

5. Diagramme de composants : il décrit le système modélisé sous forme de composants réutilisables et met en évidence leurs relations de dépendance.

6. Diagramme de déploiement : il identifie les éléments matériels (PC, Modem, Station de travail, Serveur, etc.), leur disposition physique (connexions) et la disposition des exécutables sur ces éléments matériels. Ces diagrammes se rapprochent donc plus de la réalité physique.

Sept diagrammes comportementaux :

1. Diagramme de cas d'utilisation : c'est le premier diagramme du modèle UML. Il représente la structure des grandes fonctionnalités nécessaires aux

utilisateurs du système, et il permet de voir les actions des acteurs sur le système. Ainsi, il nous montre les interactions fonctionnelles entre les acteurs et le système à l'étude.

2. Diagramme global des interactions : il permet d'établir un lien entre les diagrammes de cas d'utilisation et les diagrammes de classes, ils montrent comment des objets communiquent pour réaliser une certaine fonctionnalité. Ils apportent ainsi un aspect dynamique à la modélisation du système. Pour produire un diagramme d'interaction, il faut focaliser son attention sur un sous-ensemble d'éléments du système et étudier leur façon d'interagir pour décrire un comportement particulier.

3. Diagramme de séquence : ce diagramme montre la séquence verticale chronologique des messages passés entre objets au sein d'une interaction. En d'autres termes, il nous permet de voir les messages échangés entre les lignes de vie chronologiquement.

4. Diagramme de communication : il montre la communication entre les objets dans le plan au sein d'une interaction.

5. Diagramme de temps : il fusionne les diagrammes d'états et de séquence pour montrer l'évolution de l'état d'un objet au cours du temps.

6. Diagramme d'états-transitions : ce diagramme sert à représenter des automates d'états finis sous forme de graphes d'états, et reliés par des arcs orientés qui décrivent les transitions. Les diagrammes d'états-transitions permettent de décrire les changements d'états d'un objet ou d'un composant en réponse aux interactions avec d'autres objets/composants ou avec des acteurs.

Par ailleurs, un état se caractérise par sa durée et sa stabilité, il représente une conjonction instantanée des valeurs des attributs d'un objet. Par contre, une transition représente le passage instantané d'un état vers un autre. Alors, une transition est déclenchée par un événement, et peut aussi être automatique lorsque nous ne spécifions pas l'événement qui la déclenche.

Effectivement, le diagramme d'états-transitions est le seul diagramme, de la norme UML, à offrir une vision complète et non ambiguë de l'ensemble des comportements de l'élément auquel il est attaché. En effet, un diagramme d'interaction n'offre qu'une vue partielle correspondant à un scénario sans spécifier comment les différents scénarios interagissent entre eux.

Par contre, la vision globale du système n'apparaît pas sur ce type de diagramme puisqu'ils ne s'intéressent qu'à un seul élément du système indépendamment de son environnement.

7. Diagramme d'activité : il permet de mettre l'accent sur les traitements. Ces diagrammes sont donc particulièrement adaptés à la modélisation du cheminement de flots de contrôle et de flots de données. Ils permettent ainsi de représenter graphiquement le déroulement d'un cas d'utilisation.

En plus, le diagramme d'activité est un diagramme d'états-transitions simplifié pour lequel les états se réduisent à de simples actions ou activités et dont les transitions se déclenchent automatiquement avec éventuellement des gardes.

Par contre, la différence principale entre les diagrammes d'interaction et les diagrammes d'activités est que les premiers mettent l'accent sur le flot de contrôle d'un objet à l'autre, tandis que les seconds insistent sur le flot de contrôle d'une activité à l'autre.

En conséquence, ces treize types de diagrammes d'une utilité variable selon les cas ne sont pas nécessairement tous produits à l'occasion d'une modélisation. Pratiquement, les plus utiles sont les diagrammes d'activités, de cas d'utilisation, de classes, d'objets, de séquence et d'états-transitions. En outre, les diagrammes de composants, de déploiement et de communication sont surtout utiles pour la maîtrise d'œuvre à qui ils permettent de formaliser les contraintes de la réalisation et la solution technique.

3.4. Extensions et lacunes d'UML

D'après la littérature, UML a été appliqué à plusieurs types de systèmes hétérogènes. Par exemple, les auteurs (**Nguyen et al., 2004**) ont présenté une extension au diagramme de classe et au diagramme d'états d'UML permettant de concevoir un système temps-réel. De plus, les auteurs (**Carr et al., 2004**) ont utilisé UML pour modéliser des circuits analogiques en empruntant une approche systémique. Ils ont utilisé le diagramme de classes pour décrire la structure du système, les composants et leurs interconnexions afin de réaliser un mappage entre les ports dans le code VHDL-AMS pour permettre la description du matériel et du logiciel sur une même plateforme. En outre, une application d'UML, pour concevoir un système d'amortisseurs d'un camion, a été exposée dans (**Dudra, 2003**). Par ailleurs, d'autres travaux ont introduit dans UML des modèles pour la vérification tel que les réseaux de Pétri, et également des modèles exécutables de systèmes mécatroniques.

Ainsi, l'ensemble de ces expériences ont montré les possibilités d'UML et son utilité en tant que langage de base des systèmes complexes et/ou hétérogènes.

Contrairement, et malgré toutes ces extensions d'UML, de nombreuses tentatives d'utilisation d'UML en IS ont en effet montré que certains besoins devaient être satisfaits pour en faire un langage efficace pour les ingénieurs système, par exemple : (**Roques, 2009**)

- ➡ Besoin de décrire les exigences directement dans le modèle UML, et d'en assurer la traçabilité vers la conception.

- ➡ Besoin de représentation des flux continus (matière, fluides, énergie,...), des équations physiques, des contraintes, des éléments non-logiciels et d'en spécifier le type (mécanique, circuit, hydraulique, câblage, capteur...), des attributs de

performance, des attributs physiques et non comportementaux, des allocations logique/physique, structure/dynamique, etc.

➡ Besoin des éléments de modélisation explicites pour représenter des entrées/sorties physiques.

➡ Besoin de spécifier les événements et coupler les événements avec les actions.

Dans la littérature, beaucoup de tentatives ont été développées durant ces vingt dernières années pour répondre à ces besoins comme UML/RT, UML/PNO et des profils UML ont été définis tels que MARTE, OMEGA, TURTLE et SysML.

Par conséquent, UML reste toujours la barrière psychologique du vocabulaire orienté informatique : classe, objet, héritage, etc. De plus, UML n'est pas suffisant pour supporter un processus d'ingénierie système de systèmes complexes. Ainsi, pour obtenir un ensemble complet de modèles permettant de supporter tout le processus de conception et vu les caractéristiques de ces différents profils UML, nous avons préféré utiliser SysML qui est présenté avec plus de détails dans la section suivante.

4. *Le langage SysML*

L'ingénierie système est une approche multidisciplinaire pour développer des solutions aux problèmes de systèmes complexes. En plus, l'augmentation continue de la complexité des systèmes exige plus de pratiques formalisées d'ingénierie système. En réponse à cette demande, ainsi qu'avec des progrès importants de la technologie informatique, l'IS a proposé une transition fondamentale d'une approche basée sur des documents à une approche basée sur un modèle. Ainsi, cette approche à base de modèles peut aider à gérer la complexité, améliorer la qualité de conception et améliorer la communication entre l'équipe de développement. Donc, un langage de modélisation standard et robuste est une réponse parfaite à la complexité des systèmes.

Le langage de modélisation des systèmes complexes (SysML : Systems Modeling Language) est l'un de ces langages de modélisation à usage général qui appuie la spécification, la conception, l'analyse et la vérification de systèmes prenant en considération les matériels, les logiciels, les données, le personnel, les procédures et les équipements. SysML est un langage de modélisation graphique avec une base sémantique pour représenter les exigences, le comportement, la structure et les propriétés du système et de ses composants. De plus, il est destiné spécifiquement aux systèmes industriels tels que l'aérospatiale, l'automobile, les systèmes de production, etc. SysML a ainsi vu le jour en tant qu'extension du langage orienté-objet UML pour couvrir toutes les étapes de conception de systèmes complexes et hétérogènes (Friedenthal *et al.*, 2012).

4.1. Origines de SysML

Comme nous avons présenté dans la section précédente, le langage UML a été développé à la fin des années 90 pour répondre aux besoins des ingénieurs logiciel afin d'unifier et généraliser les pratiques de développement. Depuis son adoption en 1997 par l'OMG, UML est devenu extrêmement populaire auprès de la communauté du logiciel et devient de nos jours le seul langage de modélisation graphique largement diffusé dans ce domaine. A partir de 2003, la version 2.0 d'UML s'ouvre à l'IS en proposant de s'appliquer non plus aux « systèmes logiciels » mais aux systèmes en général. De nombreuses adaptations du langage pour des domaines spécifiques ont vu le jour grâce aux possibilités de stéréotypage et de création de « profils » supportés par la spécification UML 2.0 (**Hause, 2006**).

Néanmoins, UML n'a pas rencontré le succès prévu auprès de la communauté des ingénieurs système, principalement à cause de ses notations trop orientées logiciel et d'un manque d'expressivité pour des problématiques spécifiques aux systèmes physiques (**Willard, 2007**).

L'origine de SysML est due au conseil international d'ingénierie système (INCOSE : International Council on Systems Engineering) et au groupe de gestion d'objets (OMG : Object Management Group) qui ont organisé le groupe de travail « Model Driven System Design » créé en janvier 2001 pour personnaliser le langage UML afin d'être adapté aux applications de l'IS. En réponse l'INCOSE et l'OMG ont développé le SysML adopté en 2007 (**OMG, 2007**) et ont mis à jour dans une version 1.1 en novembre 2008 (**OMG, 2008**). En effet, SysML est construit comme un sous-ensemble des notations UML, augmenté de nouveaux artefacts destinés à traiter des spécificités de l'IS. La version officielle de SysML est 1.2, lancée en juin 2010 (**OMG, 2010**).

Ainsi, SysML supporte la spécification, l'analyse, la conception, la vérification d'un grand nombre de systèmes complexes. Ces systèmes peuvent inclure des systèmes physiques, des systèmes d'information, des processus, des personnes ou des biens. SysML est donc constitué d'une partie commune avec UML et d'une partie spécifique correspondant au profil SysML, comme illustré par la Figure 1.3.2.

Cependant, des défis majeurs étaient devant ce nouveau langage SysML (**Willard, 2007**). Par exemple, trouver une large acceptation dans la communauté de l'IS, offrir rapidement une standardisation stable et disposer d'une large gamme d'outils supportant son utilisation.

En conséquence, la spécification SysML a été mise en place pour répondre au mieux à ces challenges et aux besoins de systèmes complexes.

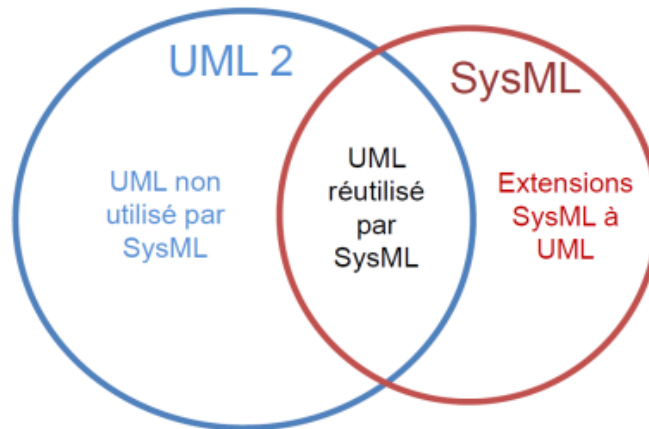


Figure 1.3.2. Relation entre SysML et UML 2.

4.2. Critères de SysML

Quand SysML a été introduit, Friedenthal et Burkhart (**Friedenthal et al., 2003**) ont proposé 9 critères que ce langage SysML devrait satisfaire :

- ✓ **Facilité d'utilisation** : le langage doit pouvoir être utilisé et compris par un large public.
- ✓ **Non-ambiguïté** : SysML doit être basé sur des sémantiques bien définies avec une notation non-ambiguë, donnant lieu à un ensemble consistant de vues de modèles qui adhèrent à des règles de bonnes constructions.
- ✓ **Précis** : le langage doit spécifier la sémantique qui peut être transformée en une représentation formelle basée sur les mathématiques. Ce critère doit faciliter l'exécution de la spécification et des modèles à tout niveau hiérarchique afin de valider les besoins et vérifier que le modèle satisfait bien à ces besoins.
- ✓ **Complet** : le langage SysML doit supporter l'expression de tous les détails émanant de la modélisation de systèmes, de l'analyse à la spécification, la conception et la vérification.
- ✓ **Adaptatif/taille du système** : ce langage doit supporter l'abstraction, l'élaboration et le raffinement (généralisation/spécialisation, décomposition, collection, vues multiples, ...) pour fournir des solutions adaptées à la modélisation de systèmes complexes.
- ✓ **Adaptatif/domaine** : SysML doit fournir les moyens d'extension vers des domaines spécifiques (aérospatial, télécommunications, automobile, ...)
- ✓ **Evolutif** : le langage doit permettre le changement, l'évolution et supporter la compatibilité avec les versions antérieures.

✓ **Echange de modèles et de diagrammes** : ce langage doit supporter l'échange d'informations et de diagrammes pour faciliter l'échange des modèles entre outils.

✓ **Indépendant du processus et de la méthode** : SysML doit pouvoir supporter des processus d'ingénierie système et ne pas contraindre un choix de processus ou méthode.

4.3. Présentation synthétique de SysML

Le but des créateurs de SysML était de proposer aux ingénieurs système un langage de modélisation à la fois simple et suffisamment expressif. Le langage SysML utilise l'approche Orientée Objet et constitue un langage de modélisation graphique sans a priori technologique sur le système modélisé. Il est conçu à la fois pour aider à l'analyse, la spécification, la conception, la vérification et la validation des systèmes complexes (OMG, 2010). Effectivement, ce langage offre la possibilité de modéliser des composants physiques ou logiciels, des acteurs humains, des flux de données, des procédures et tout ce qui peut composer un système.

Alors, SysML permet de modéliser les systèmes, leurs composants et la majorité des entités, il fournit des artefacts modélisant la structure, les interconnexions, les comportements basés sur des fonctions, les états, les contraintes, les exigences en précisant leurs relations entre elles et leur affectation à la structure et la déclaration de leur méthode de test, et en plus les allocations entre comportement/structure et exigences/contraintes.

Au niveau de l'architecture, le langage SysML est fortement inspiré de la version 2 d'UML. SysML réutilise de nombreux grands types de diagrammes UML tels que les diagrammes de cas d'utilisation, de séquence, d'états et de packages. Dans d'autres cas, des diagrammes sont modifiés de sorte qu'ils soient compatibles avec les extensions SysML tels que les diagrammes d'activité, de définition de blocs et de blocs internes. Deux nouveaux types de diagrammes ont été ajoutés à SysML, à savoir le diagramme d'exigences et le diagramme paramétrique.

Ainsi, le langage SysML s'articule autour de neuf types de diagrammes, chacun d'eux étant dédié à la représentation des concepts particuliers d'un système. Ces types de diagrammes sont répartis par l'OMG en trois grands groupes (Roques, 2009), (OMG, 2010).

La taxonomie diagramme SysML est illustrée à la Figure 1.3.3. Cette taxonomie est un exemple de la façon d'organiser les diagrammes SysML.

Quatre diagrammes comportementaux (étude dynamique) :

1. Diagramme d'activité : il montre l'enchaînement des actions et décisions au sein d'une activité complexe. Ce diagramme représente les flots de données et de

contrôle entre les actions. Il est utilisé majoritairement pour l'expression de la logique de contrôle et d'entrées/sorties.

2. Diagramme de séquence au niveau système : ce diagramme décrit chronologiquement les échanges de messages au sein d'un système. Ainsi, il montre la séquence verticale des messages passés entre éléments (lignes de vie) au sein d'une interaction. Ce diagramme est indispensable pour documenter les cas d'utilisation et la description textuelle car il permet de communiquer facilement avec les utilisateurs. En plus, il est recommandé de compléter la description textuelle par un ou plusieurs diagrammes de séquences SysML. Nous considérons le système entier comme une boîte noire et le représentons par une seule ligne de vie.

3. Diagramme d'états : il montre les différents états, transitions possibles entre eux et les actions que le système ou ses parties réalisent en réponse aux événements.

4. Diagramme de cas d'utilisation (UC : Use Case) : ce diagramme fournit une description de haut niveau des fonctionnalités du système. Il montre les interactions fonctionnelles entre les acteurs extérieurs au système et les fonctionnalités du système à l'étude.

Un diagramme transverse :

SysML a ajouté aux diagrammes du langage UML un diagramme d'exigences qui représente simplement les exigences du système et leurs relations.

Ce diagramme d'exigences est la principale extension apportée par SysML, il capture les hiérarchies d'exigences, ainsi que leurs relations de dérivation, de satisfaction, de vérification et de raffinement. Ces relations fournissent la capacité de relier les exigences les unes aux autres, ainsi qu'aux éléments de conception et aux cas de tests. Ce diagramme permet de représenter graphiquement les exigences dans le modèle et surtout de les relier ensuite aux éléments structurels ou dynamique de la modélisation, ainsi qu'à des exigences de niveau sous-système ou équipement. Les exigences servent aussi à établir un contrat entre le client et les réalisateurs du futur système.

Quatre diagrammes structurels (étude statique) :

1. Diagramme de définition de blocs (BDD : Block Definition Diagram) : le bloc est une deuxième extension importante qui est défini en tant que stéréotype de la classe définie dans UML2 et chaque bloc est une unité modulaire. Ce diagramme montre les briques de base statiques : blocs, compositions, associations, attributs, opérations, généralisations, etc. Il peut représenter un système complet, un sous-système ou un composant élémentaire. Les blocs sont décomposables. Le BDD décrit la hiérarchie du système et les classifications système/composant.

2. Diagramme de bloc interne (IBD : Internal Block Diagram) : il est aussi une extension apportée par SysML. Il montre la structure interne d'un élément statique complexe en termes de parties, ports et connecteurs.

3. **Diagramme paramétrique** : c'est un nouveau type de diagrammes. Il représente les contraintes du système, les équations qui le régissent. Il s'agit d'une spécialisation du diagramme de bloc interne où les seuls blocs utilisables sont des contraintes entre paramètres permettant de représenter graphiquement des équations et des relations mathématiques. Ce nouveau diagramme fournit ainsi un support précieux pour les études d'analyse système.

4. **Diagramme de packages** : il montre l'organisation logique du modèle et les éventuelles relations entre packages.

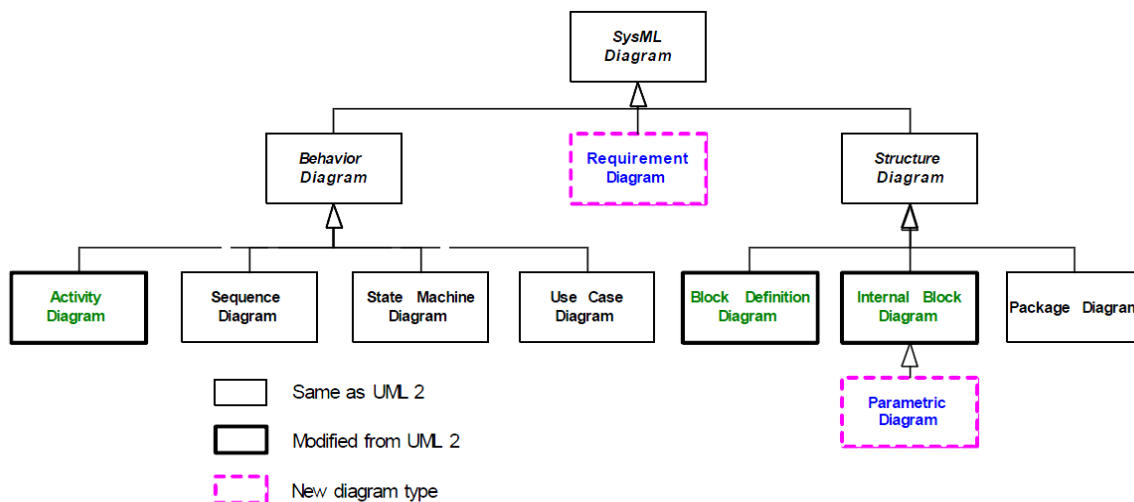


Figure 1.3.3. Diagrammes de SysML.

Enfin, un modèle SysML est principalement caractérisé par ce qui suit (OMG, 2008) :

- **Soumis à des exigences** : SysML propose des artefacts pour organiser, gérer et spécifier des exigences.
- **Composé** : les systèmes sont modélisés en termes d'associations de composants. Le modèle est organisé en répertoires.
- **Hiérarchisé** : le modèle peut être organisé en répertoires dénotant le niveau de granularité des descriptions. Les artefacts peuvent posséder des relations de filiation.
- **Interopérable** : les modèles SysML sont destinés à être échangeables et partageables entre différents outils.

Récemment, de nombreux travaux de recherche considèrent SysML le langage le mieux adapté à l'IS. Par exemple, SysML a été utilisé pour concevoir des systèmes mécatroniques (Turki, 2008), pour analyser la sûreté de fonctionnement des systèmes complexes en phase de conception (David, 2009), et également pour spécifier et modéliser l'architecture d'un système interactif d'aide à la conduite (Dobre, 2010).

5. Conclusion

Dans ce chapitre, nous avons cité quelques outils traditionnels de la modélisation d'un système, l'évolution de ces outils est détaillée et illustrée en Annexe A. Ces techniques traditionnelles sont limitées par leur expressivité ainsi que par la difficulté de leur intégration avec les exigences système. En d'autres termes, ces outils traditionnels ne s'intègrent pas au sein d'une vraie démarche cohérente de modélisation d'un système évolué dans sa globalité, ce que pourrait permettre le langage UML ou SysML.

Ensuite, nous avons introduit le langage UML comme une réponse aux lacunes des méthodes traditionnelles. En plus, UML est devenu le langage de la modélisation pour les ingénieurs logiciels. Néanmoins, UML n'a pas rencontré le succès prévu auprès de la communauté des ingénieurs système.

Finalement, nous avons présenté et décrit le langage le plus récent de modélisation de systèmes complexes SysML comme une réponse aux besoins des systèmes complexes et d'ingénieurs système. Nous avons choisi d'utiliser SysML comme langage de caractérisation et de modélisation d'un système complexe qui est dans notre cas un système de production reconfigurable et agile (SPRA). En effet, SysML nous permet de modéliser les composants de ce système, de montrer leurs définitions et leurs relations dans le modèle, de gérer les exigences liées à la reconfiguration et de montrer leurs interactions dans différents contextes. Donc, le langage SysML nous paraît le langage de modélisation le plus adapté à ces contraintes.

Nous présenterons dans le chapitre suivant les architectures de pilotage d'un système de production qui vont nous aider à piloter et superviser ce nouveau type de système de production.

Chapitre 4

Pilotage des systèmes de production reconfigurables

Le pilotage d'un système de production reconfigurable consiste à organiser les ressources disponibles, dans le temps et dans l'espace, afin d'atteindre les objectifs désirés. L'autonomie dans la prise de décision au niveau opérationnel, la coopération des composantes du système entre elles, l'agilité, la modularité et la tolérance aux perturbations apparaissent comme les qualités indispensables d'un système de pilotage compatible avec les systèmes de production reconfigurables.

Ce chapitre est consacré à la présentation des différents niveaux de pilotage d'un système de production tels que les niveaux de la planification, de l'exécution, de la commande et de l'opération. Ensuite, les architectures de pilotage référencées classiquement en fonction de leur topologie (centralisée, hiérarchisée, hétérarchisée, décentralisée, hybride...) seront présentées. Enfin, une comparaison entre ces différentes architectures sera réalisée en fonction de leurs caractéristiques, avantages et inconvénients.

Sommaire

1. Introduction	87
2. Niveaux de pilotage d'un système de production reconfigurable.....	88
3. Architectures de pilotage d'un système de production reconfigurable.....	94
3.1. Architecture de pilotage centralisée	95
3.2. Architecture de pilotage hiérarchisée	96
3.3. Architecture de pilotage hétérarchisée.....	97
3.4. Architecture de pilotage décentralisée (isoarchique).....	98
4. Comparaison entre les différentes architectures de pilotage.....	103
5. Conclusion	105

1. Introduction

Un système de production est généralement constitué d'une partie commande et d'une partie opérative. La partie commande contient la logique de fonctionnement du processus, elle envoie des ordres à la partie opérative qui les exécute à l'aide d'actionneurs. La partie opérative renvoie ensuite des informations, à partir de capteurs, à la partie commande qui gère également le dialogue avec l'opérateur par l'intermédiaire le plus souvent d'une Interface Homme-Machine (IHM) (**Chiron, 2009**).

Actuellement, la flexibilité, la réactivité et l'agilité sont des qualités incontournables et nécessaires pour des systèmes de production qui se trouvent confrontés à une demande variée et fluctuante avec des contraintes de qualité et de délais de plus en plus fortes. Le défi pour les entreprises est donc d'installer des outils de production modulaires et flexibles avec des systèmes de pilotage capables de les gérer. Ces derniers doivent, d'une part, s'adapter à l'hétérogénéité des équipements disponibles (automates, ordinateurs, robots, etc.) sachant que ces équipements peuvent être remplacés, supprimés ou reconfigurés en fonction des besoins. D'autre part, les systèmes de pilotage doivent être robustes face aux différents dysfonctionnements et aléas (**Kouiss et al., 2003**).

Par ailleurs, le pilotage des systèmes de production reconfigurables et agiles (SPRA) est en enjeu majeur dans le contexte de la compétition industrielle qui rend les conditions d'exploitation des systèmes de production plus contraignantes dans un environnement de plus en plus incertain. Cela induit la nécessité de piloter ces systèmes dans une recherche permanente de performance face aux multiples aléas qui peuvent survenir. Le pilotage et la supervision sont deux fonctions du système décisionnel dont les rôles sont inverses mais complémentaires. En effet, le rôle de la supervision consiste à collecter les données, traiter les informations, alimenter les bases de données, diffuser les résultats, en construisant ainsi des jeux d'informations représentant l'état actuel du système opérant. En d'autres termes, la supervision a pour objectif de contrôler l'exécution d'une opération ou d'un travail effectué sans rentrer dans les détails de cette exécution, elle recouvre ainsi l'aspect fonctionnement normal et anormal. Par contre, le pilotage consiste à prendre toutes les décisions utiles au fonctionnement du système de production, il doit donc indiquer de manière très concrète les actions à conduire à court terme ou bien en temps réel pour obtenir un fonctionnement efficace et efficient du système de production. Il s'agit alors d'indiquer qui fait quoi, quand, où, et comment, en respectant d'une part les contraintes définies aux niveaux décisionnels supérieurs et d'autre part celles inhérentes au système de production et à son environnement (**Pujo et al., 2002**).

L'objectif de ce chapitre est de présenter les niveaux de pilotage d'un système de production reconfigurable qui sont normalement les niveaux de la planification, de l'exécution, de la commande et de l'opération. Ensuite, les différentes architectures

des systèmes de pilotage seront également analysées de manière détaillée. Finalement, une comparaison entre ces différentes architectures sera réalisée dans un tableau récapitulatif au travers de l'étude de leurs caractéristiques, avantages et inconvénients. Ainsi, cette comparaison nous permettra d'évaluer l'intérêt de ces différentes architectures de pilotage pour que nous puissions proposer une architecture de pilotage convenable et prometteuse pour le développement d'un système de production reconfigurable et agile (SPRA).

2. Niveaux de pilotage d'un système de production reconfigurable

L'ISA (International Society of Automation) développe des standards qui sont ensuite repris comme normes par l'ANSI (American National Standards Institute). La norme ANSI/ISA-S95 est le standard international pour l'intégration des entreprises et des systèmes de pilotage. Cette norme se compose de modèles et de terminologie qui peuvent être utilisés pour déterminer quelles informations doivent être échangées entre les systèmes au niveau des ventes, de la finance, de la logistique des systèmes de production, de la maintenance et de la qualité. Egalement, la norme S95 se concentre sur les fonctions et les activités permettant la gestion de production. L'ambition de cette norme est de réduire les difficultés de l'intégration du système de pilotage en établissant un vocabulaire commun aux différentes stratégies de production en proposant une vue commune des problèmes et en construisant un modèle commun en définissant les structures de données. Ainsi, la norme S95 fournit une terminologie commune et un ensemble cohérent de concepts et de modèles. Cette norme sert à améliorer la communication, à réduire les coûts de la production, et aussi à éviter les erreurs associées à la mise en œuvre des interfaces entre l'entreprise et les systèmes de pilotage de la production (**ISA-S95, 2004**).

Le CIM (Computer Integrated Manufacturing) ou Production Intégrée par Ordinateur est un concept décrivant l'automatisation complète des processus de production largement utilisé dans les années 1980. Le CIM permet une gestion informatisée de la totalité des équipements de l'usine. Dans le milieu industriel, très concurrentiel, ce concept tend à améliorer les performances industrielles. Autrement dit, ce concept est appliqué de manière intégrée à toutes les activités, de la conception à la livraison et au service après-vente. Il utilise de multiples méthodes, moyens et techniques (informatiques et automatiques) pour simultanément améliorer la productivité, diminuer les coûts, respecter les dates prévues, augmenter la qualité des produits, assurer la flexibilité à un niveau local ou global du système de production et inclure tous les acteurs (**Doumeingts et al., 1995**), (**AFIS, 2009**).

La pyramide du CIM est une représentation conceptuelle comportant 5 échelons schématisant les niveaux de décision dans l'entreprise, du niveau le plus bas « capteurs/actionneurs » au niveau le plus haut « stratégie », en passant notamment par les niveaux « machines », « supervision » et « gestion de production ». Plus on

s'élève dans cette pyramide, plus le niveau de décision est important et plus la visibilité est globale (Figure 1.4.1). Un niveau supérieur décide ce qu'un niveau inférieur exécute (**ISA-S95, 2004**), (**Monassier, 2009**) :

- **Le niveau 4** (ERP : Enterprise Resource Planning) : Ce niveau représente la direction, la stratégie et la gestion de ressources de l'entreprise ;
- **Le niveau 3** (MES : Manufacturing Execution System) : Il s'agit de la gestion de production (des produits, des stocks, des approvisionnements, des clients, des commandes et de la facturation), l'ordonnancement et la planification de séries au niveau de l'atelier ;
- **Le niveau 2** (Supervision d'ensemble de postes de travail) : Il regroupe la localisation des produits en stocks, les mouvements physiques et la gestion des lots au niveau de la cellule ;
- **Le niveau 1** (Postes de travail ou machines) : Il correspond aux automatismes et IHM (Interface Homme Machine) au niveau des machines ;
- **Le niveau 0** (Terrain) : Il correspond aux capteurs, actionneurs et logique câblée.

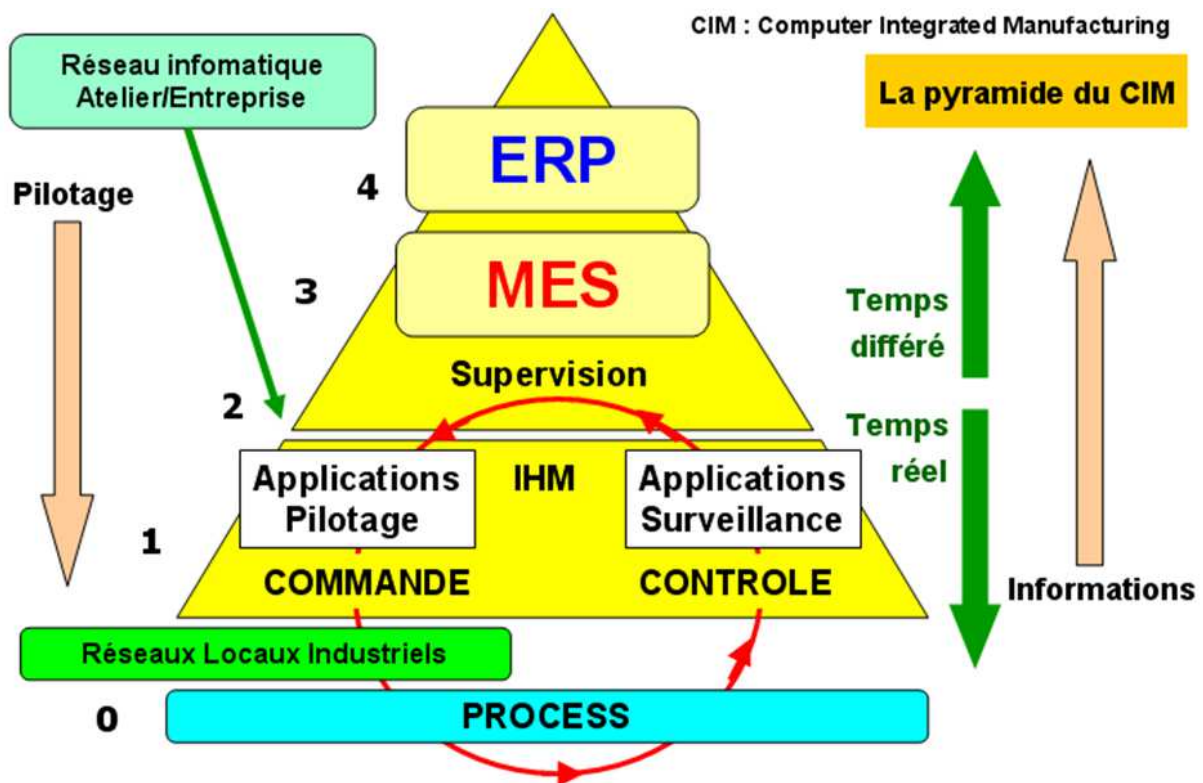


Figure 1.4.1. Pyramide du CIM.

Par ailleurs, nous trouvons dans la littérature une autre approche composée de 4 niveaux de décision dans l'entreprise (Figure 1.4.2). Cette approche est constituée

(dans le sens décroissant) des niveaux : « planification », « exécution », « commande » et « opération » (Huang, 2002), (Cardin, 2007).

Le niveau de la planification (ERP) :

Ce premier niveau contient toutes les applications permettant une vision à long terme de l'évolution du système. Diverses décompositions sont encore faites à l'intérieur de ce niveau, mais ce n'est pas l'objet de notre contribution.

Le niveau de l'exécution (MES) :

Il contient les applications permettant la gestion des produits de leur entrée dans le système à leur sortie.

Par définition, un MES (Manufacturing Execution System) est un ensemble de composants matériels et logiciels permettant la gestion et l'optimisation des activités de production du lancement de la commande aux produits finis tout en maintenant des données précises et à jour. Le MES guide, initie, répond et rapporte les événements de l'usine au moment où ils se produisent (Barkmeyer *et al.*, 1999), (Huang, 2002), (Kouiss *et al.*, 2003), (ISA-S95, 2004).

Le MES est une gamme d'outils qui ne joue pas seulement un rôle d'agent de liaison bidirectionnelle entre la planification globale de l'entreprise ERP et la commande/contrôle des équipements de production. Il a aussi pour rôle de mettre en œuvre de façon opérationnelle et optimisée, les instructions reçues de la planification, de surveiller, mémoriser et permettre d'analyser tous les événements normaux et critiques liés à celle-ci afin de rendre compte aux services fonctionnels de l'entreprise. En plus, une des tâches les plus difficiles à accomplir pour le MES est de mettre en harmonie la base de temps de l'ERP qui est du type transactionnel jour, semaine, mois, avec celle du contrôle/commande qui est du type temps réel.

Plus précisément, un MES est un système informatique dont les objectifs sont d'abord de collecter en temps réel les données de production de tout ou partie d'une usine ou d'un atelier. Ces données collectées permettent ensuite de réaliser un certain nombre d'activités d'analyse : traçabilité, contrôle de la qualité, suivi de production, ordonnancement, maintenance préventive et curative. Effectivement, son utilisation conduit à un fonctionnement plus efficace de l'atelier de production, diminue les stocks et augmente les marges.

D'autre part, le MESA (Manufacturing Enterprise Solutions Association) a défini une liste détaillée de fonctionnalités, connues les « 11 fonctions du MES » dont certaines ont un caractère opératoire tandis que d'autres concernent plutôt des fonctions transversales de l'entreprise (Lenclud, 1993), (Barkmeyer *et al.*, 1999), (Le Moigne, 1999), (ISA-S95, 2004), (<http://www.club-mes.com>, juillet 2013), (wikipédia, juillet 2013), (<http://www.actors-solutions.com>, juillet 2013) :

1. Gestion des ressources : Cette fonction suit le statut des ressources (personnel, matériels, équipements, documents, énergie, matières premières) et met à jour un historique détaillé ;

2. Ordonnancement : Il s'agit de la gestion des ordres de production en tenant compte de l'état réel de l'atelier (ressources disponibles et/ou autorisées), en prenant en considération ses performances nominales et en visant l'optimisation de son fonctionnement ;

3. Cheminement des produits et des lots (suivi des opérations) : Cette fonction dirige le flux de produits selon le plan de production et l'ordonnancement détaillé en suivant la production en temps réel, en prenant en compte les aléas de production et les parcours sur différentes lignes ou machines. De plus, cette fonction permet l'optimisation des stocks et la prise en compte de tâches non prévues (maintenance, attente,...) ;

4. Gestion des documents de l'atelier : Elle contrôle, gère et restitue diverses informations relatives aux produits dont les gammes, les procédures opératoires standards, modes opératoires, etc ;

5. Collecte et acquisition des données : Il s'agit de collecter et mettre à jour les informations de production utilisées, à travers des automates, IHM et PC de supervision, pour le suivi des produits, les historiques de production, etc ;

6. Gestion du personnel : Cette fonction permet un suivi « à la minute près » du statut du personnel (présence, fonction, disponibilité, etc.) ;

7. Gestion de la qualité : Il s'agit de mettre en œuvre de différentes méthodes de contrôle pour assurer une qualité suffisante de la production ;

8. Gestion/Pilotage des procédés : Cette fonction est totalement décrite dans les précédentes fonctions (3 et 7). Par contre, cette distinction n'est réalisée que pour préciser qu'un système différent peut être utilisé pour réaliser cette fonction ;

9. Gestion de la maintenance : Il s'agit de planifier et enregistrer des actions de maintenance préventive en termes de périodicité et de contenu, et également de déclencher et enregistrer de la maintenance curative. Donc, cette fonction historique, dirige, assure la périodicité et gère la disponibilité des outils nécessaires aux opérations de maintenance par machine et par atelier. D'autre part, la gestion de maintenance des équipements est une nécessité pour l'obtention de la certification ISO ;

10. Traçabilité ascendante et descendante du produit : Cette fonction permet une visibilité sur la disposition spatiale et temporelle de chaque opération. Elle permet de garder une trace de tous les événements importants de la production ;

11. Analyse des performances : Elle permet un suivi des résultats courants des opérations de production (utilisation/disponibilité des ressources, temps de production unitaire, conformité à l'ordonnancement prévu, etc.), et également une comparaison à l'historique et aux objectifs de performance fixés par l'entreprise, les clients ou les contraintes réglementaires. En effet, un certain nombre d'indicateurs de performance pour la production et la maintenance font l'objet de définitions quasi-normalisées.

Pratiquement, ces fonctions doivent être complétées et spécialisées selon le contexte. En effet, tout ou partie de ces fonctions peuvent être utiles selon le type d'industrie (mécanique, pharmaceutique, agroalimentaire, chimique, automobile, aéronautique, électronique...). Nous trouvons ainsi sur le marché des logiciels dédiés à un type d'industrie comme par exemple le secteur pharmaceutique.

Par conséquent, les 11 fonctions ne sont pas forcément toutes nécessaires pour une usine ou pour un secteur industriel donné. De même, cette répartition n'est pas exempte de certains « recouvrements » ou « doublons ». C'est pour ces deux raisons que chaque fonction est souvent associée à un module logiciel qui peut ou non être installé, cela traduit exactement la notion de la modularité qui est l'une des caractéristiques principales des systèmes de production reconfigurable (RMS).

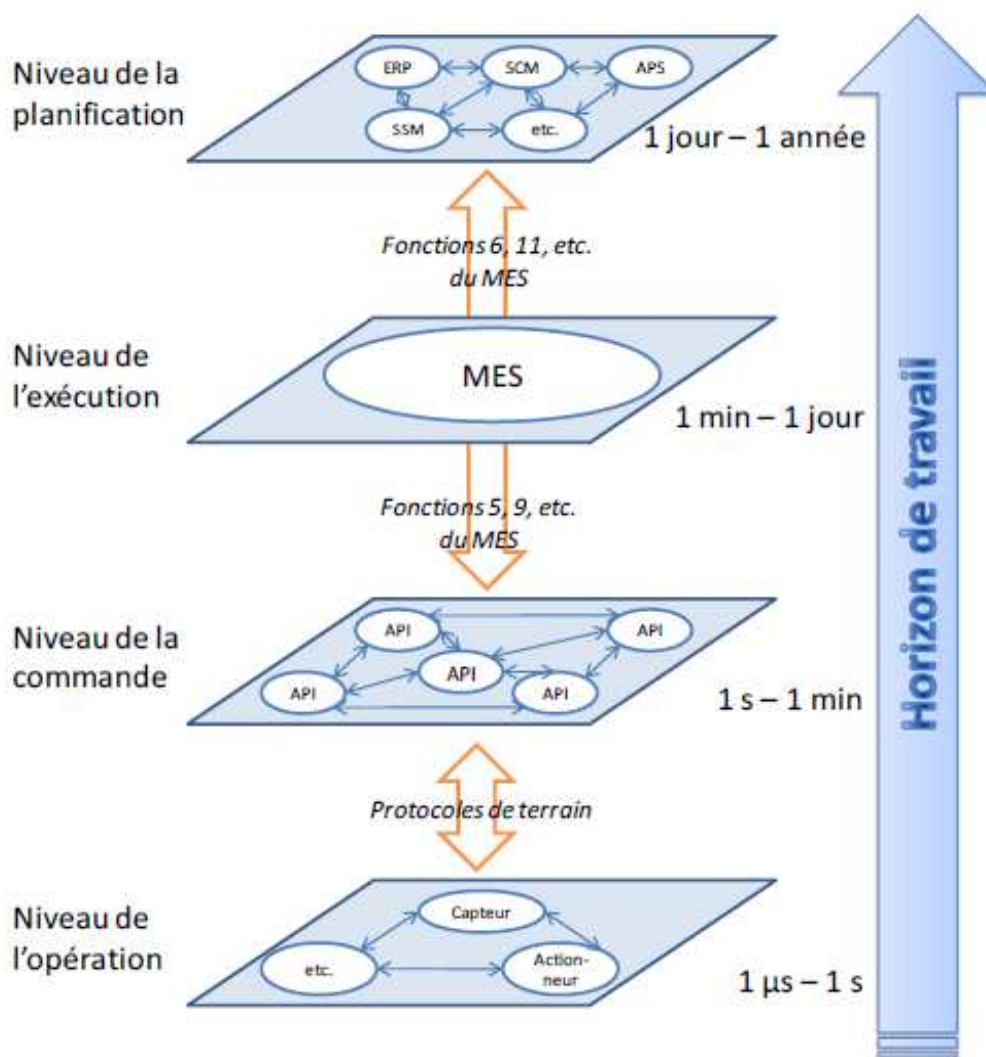


Figure 1.4.2. Niveaux de décision dans l'entreprise.

Le niveau de la commande :

Il comprend les applications et les matériels nécessaires au contrôle/commande du système de production.

Le niveau de l'opération :

Ce niveau contient tous les matériels nécessaires à la détection et à l'actuation : capteurs, actionneurs, etc.

Effectivement, la communication entre les deux derniers niveaux (commande, opération) se fait au moyen de différents réseaux industriels de terrain (Ethernet industriel, Profibus, ASi, ProfiNet, etc.). En plus, au travers des fonctions décrites ci-avant, le MES est l'outil de communication permettant un lien entre les trois premiers niveaux (planification, exécution, commande). La partie 3 de la norme S95 se concentre sur les fonctions et les activités pour la gestion des opérations de la production, cette norme définit alors les interfaces entre la commande du procédé et les autres fonctions de l'entreprise (Figure 1.4.3). Le but est de réduire les risques, les coûts et les erreurs associés avec l'implémentation de cette interface. En plus, la norme S95 essaie de réduire les efforts associés à l'implémentation de nouvelles productions (ISA-S95, 2004).

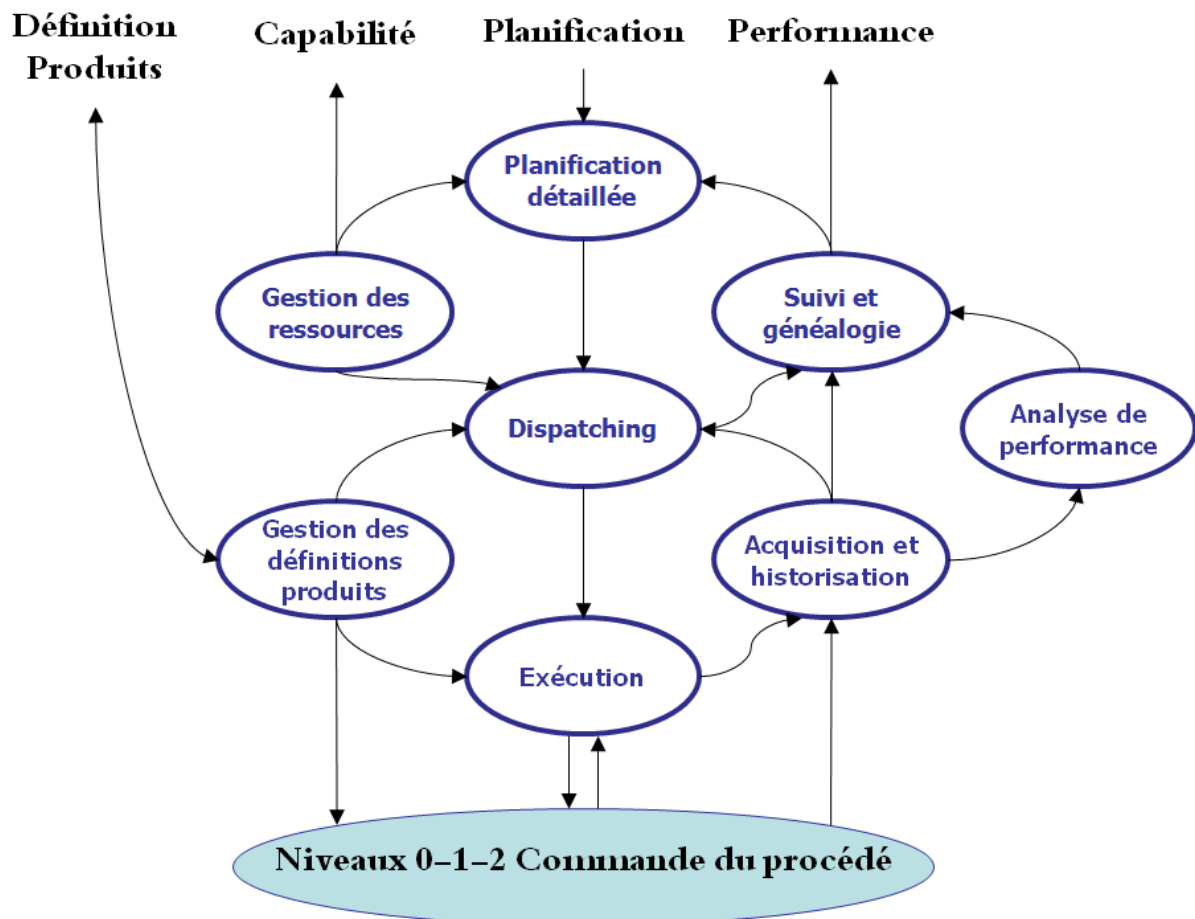


Figure 1.4.3. La modélisation opérationnelle de la gestion des opérations de la production (Norme S95).

Finalement, les deux décompositions présentées sont relativement artificielles. Les frontières entre ces différents niveaux sont souvent mal définies. Ainsi, le

concept de capteur intelligent se situe aujourd'hui entre les niveaux opération et commande. De même, l'intégration aujourd'hui est de plus en plus poussée des niveaux ERP et MES, les outils ERP intègrent de plus en plus les fonctions du MES et vis-versa.

Dans ce contexte, l'objectif de ces travaux de thèse est d'ajouter des nouvelles fonctionnalités qui servent à développer un système de production reconfigurable et agile (SPRA) se situant entre les niveaux exécution et commande.

3. Architectures de pilotage d'un système de production reconfigurable

Il est difficile d'établir une classification exhaustive de toutes les architectures de pilotage des systèmes de production. Effectivement, divers travaux de recherche se sont intéressés aux architectures des systèmes de pilotage et ont permis d'analyser leur évolution, leurs avantages et leurs inconvénients (Dilts *et al.*, 1991), (Pels *et al.*, 1997), (Pujo *et al.*, 2002), (Pujo *et al.*, 2009a), (Pujo, 2009b), (Trentesaux, 2002), (Trentesaux, 2007), (Cardin, 2007), (Trentesaux, 2009), (Darmoul, 2010), (Sallez *et al.*, 2010), (Raileanu, 2011), (Pach *et al.*, 2014).

Cependant, nous présentons un panorama des principales architectures de pilotage possibles, les plus connues et les plus habituelles (Figure 1.4.4), (Figure 1.4.5). Cette classification distingue les architectures en fonction de leur topologie : centralisée, hiérarchisée, hétérarchisée et décentralisée (isoarchique).

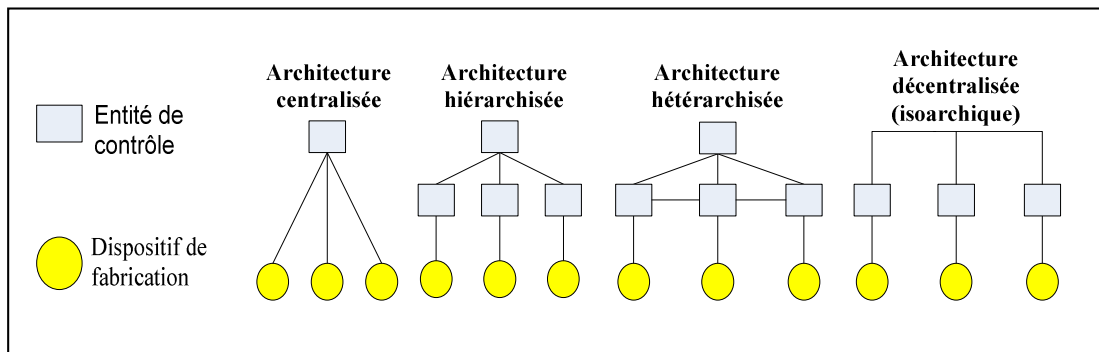


Figure 1.4.4. Architectures classiques de pilotage.

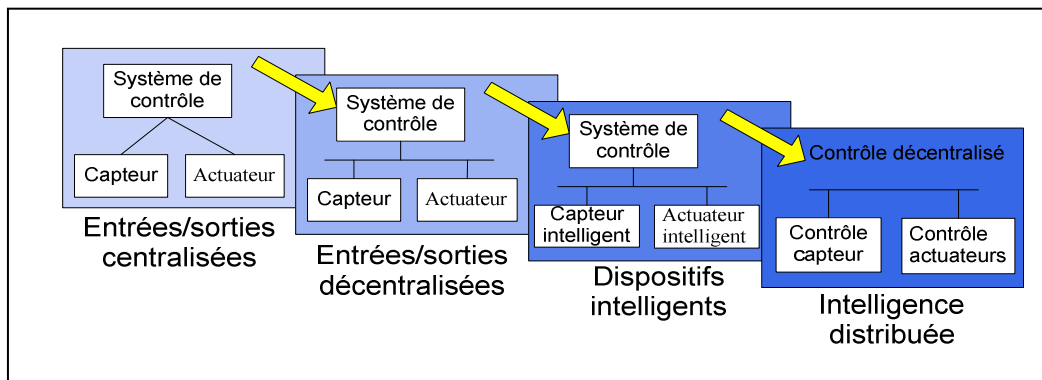


Figure 1.4.5. Evolution des architectures opérationnelles de pilotage.

Il est important de noter que cette classification peut être proposée sous autres définitions. Par exemple, l'architecture « hétérarchisée » définie dans les travaux de (Trentesaux, 2002) est proche de l'architecture « hiérarchisée modifiée ou coordonnée » proposée par (Pujo *et al.*, 2002). Egalement, dans la littérature, nous trouvons d'autres architectures comme : hybride, distribuée supervisée, etc.

3.1. Architecture de pilotage centralisée

Dans ce cas, il existe un seul point de pilotage qui prend toutes les décisions concernant les flux informationnels et décisionnels du système cible (Figure 1.4.6). Ce point central gère en temps réel les événements et réalise la synchronisation et la coordination de toutes les tâches.

Cette architecture simple (capteurs et actionneurs connectés à un seul point décisionnel) permet une meilleure gestion et une optimisation globale du fonctionnement. En effet, cela représente le principal avantage de ce type de pilotage.

Par contre, les désavantages de cette architecture sont :

- les temps de réponse de durée importante vis-à-vis de la dynamique du système lorsque le système de pilotage présente un nombre important d'entrées/sorties ;
- la difficulté d'apporter des changements dans la structure physique du système de pilotage qui n'est pas modulaire, et cela empêche principalement de maintenir correctement le système ;
- une certaine rigidité qui réduit les capacités de régulation du système ;
- cette architecture est peu robuste et ne peut donc efficacement intégrer la résistance aux perturbations : un petit problème dans le système pouvant provoquer son arrêt complet, et le système complet devient alors inutilisable jusqu'au moment de sa remise en fonction.

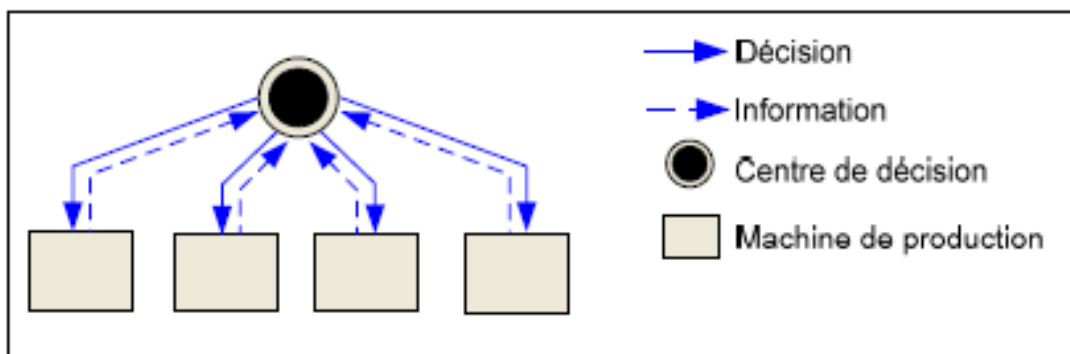


Figure 1.4.6. Architecture de pilotage centralisée.

Pour pallier aux inconvénients de cette architecture, des auteurs ont proposé l'architecture hiérarchisée suivante.

3.2. Architecture de pilotage hiérarchisée

Cette architecture est une extension de la précédente, elle est caractérisée par l'existence de plusieurs niveaux de pilotage de telle façon que chaque niveau coordonne les unités de pilotage du niveau inférieur et ce, jusqu'au niveau le plus bas. Chaque niveau a donc des relations de dépendance vis-à-vis du niveau supérieur et de dominance vis-à-vis du niveau inférieur. Dans cette architecture, chaque décision est élaborée au niveau où un problème est détecté. Les niveaux inférieurs traitent alors cette décision comme une contrainte et transmettent en retour une information de suivi au niveau supérieur (Figure 1.4.7).

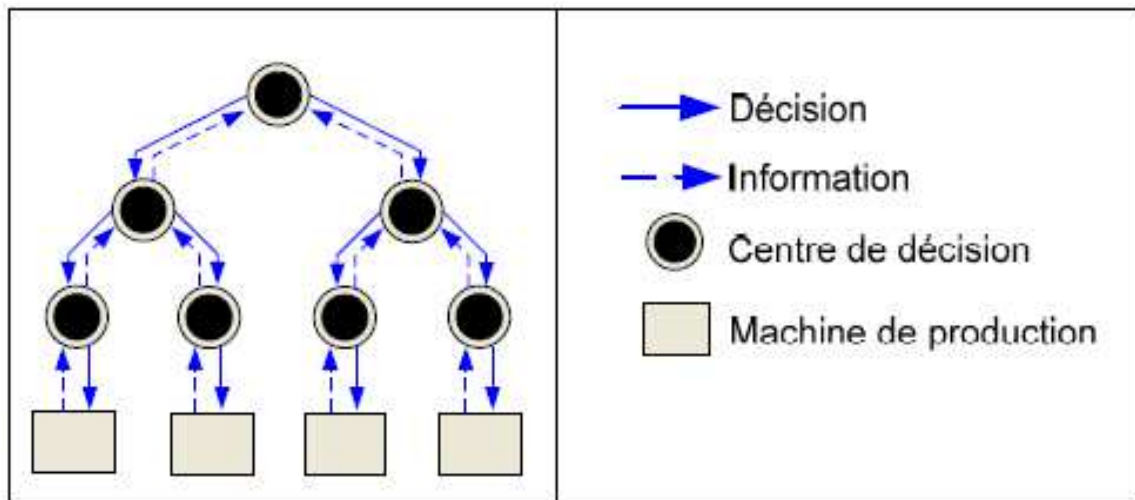


Figure 1.4.7. Architecture de pilotage hiérarchisée.

Cette architecture a été adoptée systématiquement pour les systèmes de tailles importantes jusque dans les années 1980.

Les principaux avantages de cette architecture résident dans sa prévisibilité globale, dans la réduction des problèmes de développement de programmes de contrôle, et également dans l'optimisation globale du fonctionnement du système.

Par contre, certaines réserves peuvent être formulées, notamment en termes de réaction aux perturbations, de maintenance du système de décision et d'accessibilité au système d'information. De plus, les auteurs soulignent la difficulté d'ajouter, de modifier ou de supprimer des ressources, et cela ne peut être réalisé sans l'arrêt du système et la mise à jour de tous les niveaux hiérarchiques. Autrement dit, le principal problème de cette architecture est qu'elle contient de nombreuses boucles de décisions hiérarchisées et que donc la réactivité du système est rapidement limitée. Lorsque l'environnement du système change rapidement, son adaptation et le maintien de ses performances deviennent très difficiles (Trentesaux *et al.*, 2000) et (Pach *et al.*, 2011).

En conséquence, cette approche de pilotage offre des solutions optimales mais sous réserve que des hypothèses strictes soient satisfaites (ex. absence d'ordres urgents, pas de pannes de ressources, stabilité des temps opératoires des ressources...). Mais la réalité est rarement aussi déterministe et cette approche devient rapidement inefficace lorsque le système doit faire face à un nombre d'aléas et de modifications importants qui constituent précisément l'un des objectifs principaux dans le développement des systèmes de production reconfigurables et agiles (SPRA). Donc, il nous semble que ces deux architectures de pilotage (centralisée et hiérarchisée) ne conviennent pas pour atteindre les objectifs fixés dans le cadre de notre travail de thèse.

3.3. Architecture de pilotage hétérarchisée

Il s'agit d'une architecture hiérarchisée modifiée qui permet la communication et la coordination entre les entités du même niveau. Elle permet de compléter l'architecture hiérarchisée précédente par l'existence d'une coopération par niveau (Figure 1.4.8).

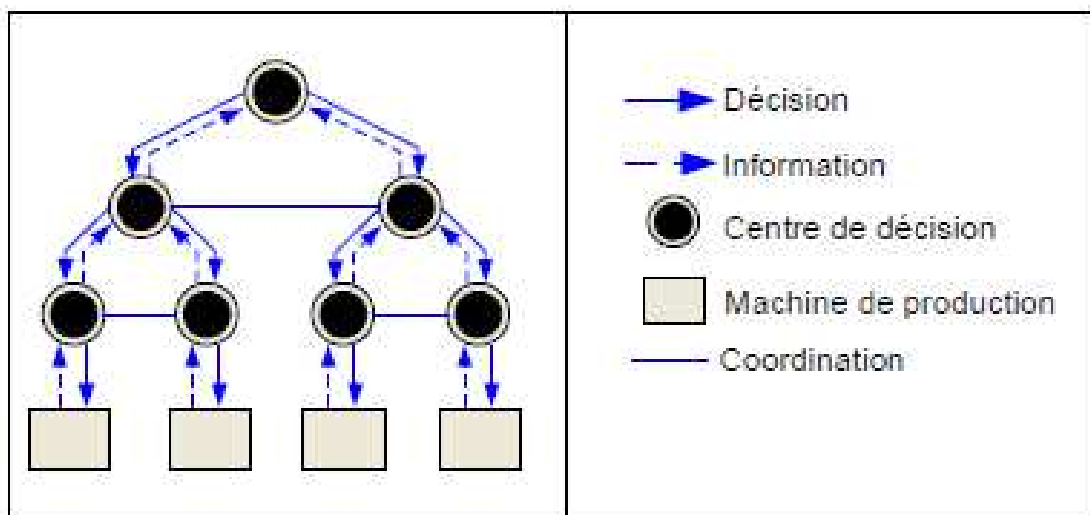


Figure 1.4.8. Architecture de pilotage hétérarchisée.

Ainsi, ce type de pilotage a été mis en œuvre par différentes approches telles que MSI (Manufacturing Systems Integration) (**Senehi *et al.*, 1994**), FACT (Factory Activity Control) (**Arentsen, 1995**), PAC++ (Production Activity Control) (**Andersson, 1997**), CIM (Computer Integrated Manufacturing) (**Quintas *et al.*, 1997**), MOSCOT (Modular shop Control Toolkit) (**Teunis *et al.*, 1998**), CHAMP (Chalmers Architecture and Methodology for Flexible Production) (**Gullander, 1999**).

L'avantage d'une telle architecture est bien évidemment sa robustesse : l'ajout d'une entité ou la suppression temporaire pour cause de panne par exemple sont

faciles à mettre en œuvre. Effectivement, cela augmente la capacité de décision locale au sein de chacun de ces niveaux, et favorise ainsi la réactivité par une meilleure intégration en temps réel des perturbations.

Par contre, cette architecture souffre de manque de visibilité globale des entités décisionnelles et à leur manque de connaissance sur le système qu'il est qualifié de « myopie ». Ainsi, dans (Trentesaux, 2009) et (Zambrano Rey *et al.*, 2014) la myopie est définie comme étant les conditions qui génèrent un manque de visibilité précise des états futurs d'un système. Ainsi, la myopie présente deux aspects (Pach *et al.*, 2011), (Pach, 2013) : la première catégorie de myopie est la myopie temporelle qui est définie comme le manque d'information que possède une entité sur le futur. La seconde catégorie de myopie est la myopie spatiale qui est définie comme le manque d'information que possède l'entité sur son environnement.

Par conséquent, l'introduction de la communication a justement pour but d'améliorer l'optimisation de ces systèmes répartis.

3.4. Architecture de pilotage décentralisée (isoarchique)

Cette architecture représente un cas particulier de l'architecture hétérarchisée. Dans cette approche, tous les centres de décision sont au même niveau fonctionnel et il n'existe pas de centre de décision de niveau hiérarchique supérieur pour coordonner l'ensemble (Figure 1.4.9).

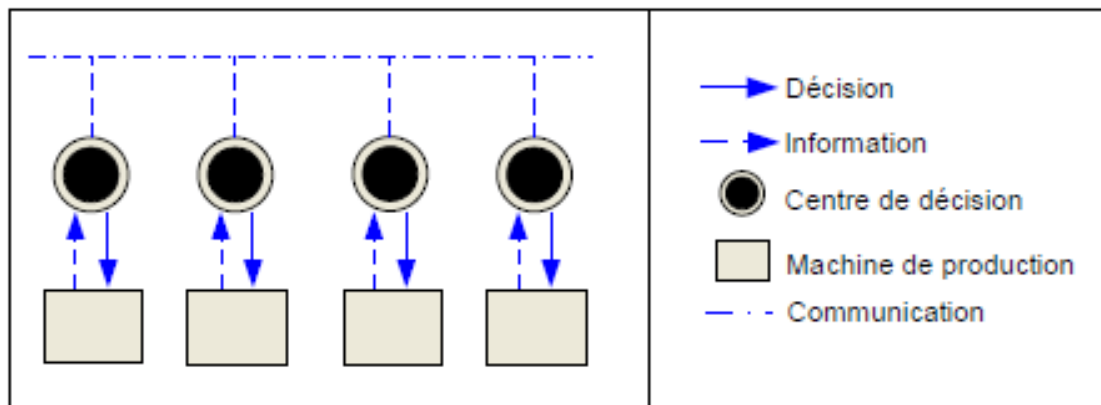


Figure 1.4.9. Architecture de pilotage décentralisée (isoarchique).

En outre, dans les travaux de (Pujo *et al.*, 2008), (Pujo *et al.*, 2009a) et (Pujo, 2009b) une architecture décentralisée est dite isoarchique lorsque ses centres de décision sont capables d'organiser eux-mêmes leur propre fonctionnement, de manière autonome, coordonnée et égalitaire, sans la direction d'aucun centre de décision de niveau hiérarchique supérieur, l'ensemble permettant le fonctionnement global du système. Cela signifie donc la même autorité pour les différents centres de

décision et une absence totale d'hierarchie. Ceci induit un certain nombre de propriétés :

- ✓ Même capacité de décision (égalité de pouvoir) ;
- ✓ Mécanismes de décision décentralisés, dupliqués sur chaque centre de décision et paramétrés à façon sur chacun ;
- ✓ Système d'interaction et de coordination à un seul niveau de communication ;
- ✓ Auto organisation du fonctionnement (pas de prédictif).

Pratiquement, cette architecture bénéficie de l'avantage d'une grande autonomie et coopération qui permettent une grande performance contre les perturbations et rendent le système facilement extensible. Par contre, l'optimisation globale est réduite parce que la prise de décision est locale, autonome et sans vue globale sur le système.

Par ailleurs, cette architecture décentralisée est proche de l'architecture distribuée (**Trentesaux, 2009**) dans laquelle les activités décisionnelles peuvent être considérées comme des activités de pilotage locales (par exemple, le pilotage d'une ressource locale). Le pilotage distribué soutient fortement l'agilité qui peut être définie comme un indicateur qualitatif car elle représente la capacité de réagir efficacement dans une courte période de temps dès l'apparition de perturbations inattendues dues aux marchés et/ou aux demandes de clients (**Leitao et al., 2008**).

Dans la littérature (**Boursbia et al., 2002**), (**Trentesaux, 2009**), (**Sallez et al., 2010**) et (**Pach et al., 2012**), de nombreux travaux de recherche proposent encore de multiples mécanismes de décision décentralisés. Par exemple, le pilotage par les systèmes multi-agents et le pilotage holonique. Également, le pilotage par le produit peut être une solution prometteuse dans certains cas (**McFarlane et al., 2003**), (**McFarlane et al., 2013**). Nous décrivons brièvement ces concepts dans les sections suivantes :

• **Pilotage par les systèmes multi-agents :**

Par définition, un système multi-agents est un système composé d'un ensemble d'agents, situés dans un certain environnement et interagissant selon certaines relations pour résoudre un problème complexe dont la résolution dépasse la capacité individuelle de chacun des agents pris à part. Un agent est une entité caractérisée par le fait qu'elle est, au moins partiellement, autonome. Ce peut être un processus, un robot, un être humain, etc.

Par ailleurs, chaque agent a des informations ou des capacités limitées de résolution de problèmes, ainsi chaque agent a un point de vue partiel et il n'y a aucune vision globale du système. Les données sont décentralisées et le calcul est asynchrone. Les propriétés les plus importantes d'un agent sont : l'autonomie, l'intelligence, l'adaptation et la coopération.

En effet, le but principal d'un système multi-agents est de faire collaborer un certain nombre d'agents afin de résoudre un problème. Ainsi, les systèmes multi-agents permettent de distribuer les problèmes sur des ensembles auto-organisés vers une gestion locale favorisant la réactivité du système. Donc, l'avènement des systèmes multi-agents peut faciliter la mise en œuvre du pilotage en temps réel (Hafri *et al.*, 2001), (Shen *et al.*, 2006), (Ali, 2009), (Leitao, 2009), (Guo *et al.*, 2009), (Trentesaux, 2009), (Sallez *et al.*, 2010), (Adhau *et al.*, 2012).

Dans la littérature, les travaux adressant le pilotage par les systèmes multi-agents ont apporté des contributions au niveau de la conception de protocoles d'interaction entre les agents et au niveau du développement de l'architecture de pilotage.

Les protocoles d'interaction définissent les mécanismes de coordination et/ou de négociation qui peuvent exister entre les agents. Dans la plupart des systèmes de pilotage à base d'agents, la négociation est utilisée pour assurer l'allocation de ressources.

Au niveau de l'architecture de pilotage, deux niveaux sont considérés : un niveau matériel proche des ressources et des processus de production : les machines, les robots, etc. Un niveau supérieur qui assure la coordination entre les ressources et les processus de production pour produire les types et les quantités de produits désirés.

Effectivement, de nombreuses applications du pilotage décentralisées ont été mises en œuvre à l'aide de structures basées sur le concept de systèmes multi-agents. Leurs résultats expérimentaux montrent que cette approche multi-agents produit des solutions efficaces et de haute qualité à court temps, qui pourrait conduire à une utilisation en temps réel de ce système de pilotage décentralisée (Trentesaux, 2009).

- **Pilotage holonique :**

Le concept holonique a été initialement introduit par l'écrivain et philosophe hongrois Arthur Koestler (Koestler, 1967) qui étudiait les organismes vivants et les organisations sociales complexes. Il a mis en évidence le fait que, dans la vie réelle, une entité, pouvant être un tout ou une partie, ne peut-être considérée comme un absolu. Il a alors introduit le concept de Holon, concaténation du terme « holos », signifiant le tout et du suffixe « on » suggérant une partie. Chaque holon répond strictement à trois conditions : être stable, avoir une capacité d'autonomie et être capable de coopérer.

Dans un système de production holonique, le holon comporte alors à la fois une dimension physique qui correspond à une réalité du système physique de production, et une dimension logique qui possède des informations et est doué d'intelligence. De plus, comme dans la définition générique de Koestler, les holons doivent obéir à des règles générales fixées par l'organisation et qui permettent d'atteindre un objectif commun.

Le concept holonique a été mis en œuvre dans plusieurs domaines tels que la planification, l'ordonnancement, la gestion de stock et d'entrepôt, et le pilotage

d'atelier. Plusieurs de ces travaux reposent sur l'architecture PROSA (Product-Resource-Order-Staff-Architecture) introduite par des chercheurs de l'Université Catholique de Louvain (**Van Brussel et al., 1998**). PROSA est une architecture holonique de référence basée essentiellement sur trois types de holons : produit, ressources et les ordres. Le personnel peut être ajouté comme un holon pour aider les holons de base avec des connaissances spécialisées. Cette architecture résultant réduit la complexité de l'intégration des nouveaux composants et permet ainsi une reconfiguration facile du système.

Par contre, les holons n'ont qu'une vue partielle de l'état du système : le holon produit contient le processus de fabrication du produit, le holon ressources contient une partie physique (les équipements de production) et une partie de traitement des informations (les méthodes d'allocation des ressources), le holon ordres représente une tâche dans le système de production, il est ainsi responsable de l'exécution de la tâche assignée dans les délais impartis. Les interactions entre ces trois types de holons assurent la gestion des connaissances de la production. Le holon personnel n'a pas de pouvoir de décision, mais peut proposer une solution globale pour résoudre un problème en fonction des données qui sont transmises par les holons de base qui restent responsables de la prise de décision finale. Cependant, l'approche isoarchic peut être mise en œuvre par le concept holonic (**Blanc et al., 2008**), (**Cardin et al., 2009**), (**Trentesaux, 2009**), (**Pujo et al., 2009a**), (**Sallez et al., 2010**).

- **Pilotage par le produit :**

Un produit est dit intelligent si il est porteur d'informations et capable de faire varier son comportement en réponse aux différentes situations et exigences. Celui-ci est représenté sous la forme d'une partie physique et d'une partie informationnelle. Selon les auteurs (**McFarlane et al., 2003**), (**Meyer et al., 2009**) un produit intelligent se doit d'avoir tout ou partie de ces cinq caractéristiques :

1. Il possède une identification unique ;
2. Il est capable de communiquer efficacement avec son environnement ;
3. Il peut retenir et conserver des données le concernant directement ;
4. Il utilise un langage pour communiquer ses besoins et ses possibilités ;
5. Il est capable de participer à la prise de décision concernant son devenir.

L'idée d'intégrer l'intelligence et le contrôle au niveau du produit a donné lieu à des travaux dans le domaine du pilotage de systèmes de production (**McFarlane et al., 2003**), (**Trentesaux, 2009**). Dans ce domaine, les nouvelles technologies d'auto-identification (Auto-ID) telles que l'identification radio fréquence RFID (Radio Frequency IDentification) et les réseaux de capteurs sans fils ont rendu possible la traçabilité des produits à travers tout le système de production. Egalement, l'intérêt de l'utilisation de cette technologie est de rendre l'adaptation du processus de

production au produit facile et de stocker toutes les informations nécessaires sur la même étiquette. Ainsi, la gestion des palettes mixtes contenant plusieurs références de produits sera encore plus facile.

Effectivement, le développement des systèmes contrôlés par le produit permet de créer des systèmes de production partiellement voire totalement réactifs (**Morel et al., 2003**), (**Pannequin et al., 2006**).

Dans la littérature, des travaux de recherche basés sur le pilotage par le produit ont été menés pour améliorer les performances du pilotage opérationnel des systèmes de production flexibles (**Sallez et al., 2009**) et pour aller vers une plus grande personnalisation des produits (**Meyer et al., 2009**).

Plus précisément, dans le domaine des systèmes de production reconfigurables, une approche basée sur le principe de line-less a été déjà proposée (**Ueda et al., 2001**) :

Le concept Line-less :

Il s'agit d'une approche pour la conception de systèmes de production line-less qui peuvent faire face à la complexité de la production en utilisant les idées d'auto-organisation, l'évolution et l'apprentissage. Dans ce cas, tous les éléments de production peuvent se déplacer librement dans l'atelier en utilisant l'auto-organisation dans le but de s'adapter aux fluctuations telles que la diversité des demandes de production et le dysfonctionnement de machines.

Le concept Line-less est proposé dans le but de réaliser des systèmes de production entièrement flexibles ou bien reconfigurables afin de s'adapter aux changements dynamiques dans l'environnement de production. Ainsi, les processus de production continuent alors que tous les éléments autonomes se déplacent librement sur le terrain.

Les principaux avantages de ce concept résident dans les points suivants :

- Variété élevée de produits : divers types de produits peuvent être produits sur la même ligne de production ;
- Robustesse des dysfonctionnements : même si une ressource est en panne, les autres ressources ne sont pas influencés beaucoup parce que cette ressource doit être réparée ou calibrée individuellement ;
- Reconfiguration facile : quand un nouveau produit est demandé, de nouvelles ressources peuvent être ajoutées sans arrêter les processus de production. En outre, la nouvelle ressource peut être apprise en ligne.

Par contre, les inconvénients suivants peuvent être présents :

- Nécessité d'un atelier de production de grande taille afin que toutes les ressources puissent se déplacer librement sur le terrain ;

- Augmentation du coût: parce que les nouvelles ressources mobiles devraient être développées et mises en place.

Il est important de noter que le pilotage de ce système est généralement difficile en raison de grand nombre des degrés de liberté du système. Pour cela, une approche auto-organisation a été ensuite introduite et intégrée avec le concept Line-less.

Enfin, selon les auteurs (**Ueda *et al.*, 2001**) un système de production Line-less en utilisant l'auto-organisation a été adopté pour un procédé de soudage du châssis de soins comme un banc d'essai, et une simulation par ordinateur a été effectuée.

4. Comparaison entre les différentes architectures de pilotage

Dans la section précédente, nous avons présenté les principales architectures de pilotage en focalisant sur les caractéristiques, les avantages et les inconvénients de chaque architecture. L'évolution des architectures a été accompagnée par une augmentation correspondante de l'autonomie des coopératives de composants de pilotage, par une réduction de l'utilisation de l'information globale et par un relâchement général ou une suppression des relations maître/esclave. Effectivement, les avancées dans la technologie informatique et les communications ont fortement influencé ces changements.

Généralement, l'environnement d'un système de production impose des exigences élevées sur l'architecture de pilotage. Ainsi, la simplicité, la robustesse, l'optimisation globale du fonctionnement du système, la fiabilité, la prévisibilité globale, l'autonomie, la coopération, l'intelligence, l'adaptabilité, la reconfigurabilité (modularité, extensibilité,...) et la réactivité ont été mises en évidence comme les objectifs essentiels de la conception d'architectures de pilotage. Cet ensemble d'objectifs est devenu de plus en plus intériorisé au sein des architectures ce qui les a poussées à évoluer comme déjà présentées dans la section précédente. Le Tableau 1.4.1 récapitule les avantages et les inconvénients les plus importants de chaque architecture de pilotage.

	Robuste	Optimisation globale du fonctionnement	Prévisibilité globale	Autonomie	Coopération par niveau	Reconfigurabilité	Réactivité	
Architecture centralisée	Peu robuste (-)	Oui (++)	Oui (++)	Non (--)	Non (--)	Non (--)	Non (--)	
Architecture hiérarchisée		Oui (+)		Peu autonomie (-)			Limitée (-)	
Architecture hétérarchisée		Non (--)		Oui (+)			Oui (++)	
Architecture décentralisée (isoarchique)								
Architecture distribuée	Oui (++)	Réduite (-)	Non (-)	Oui (++)	Oui (++)		Oui (+)	
Pilotage par les systèmes multi-agents								
Pilotage holonique								
Pilotage par le produit	Le concept Line-less						Oui (++)	
							Oui (+)	

Tableau 1.4.1. Tableau récapitulatif des différentes architectures de pilotage.

Finalement, ce tableau permet d'évaluer l'intérêt de ces différentes architectures de pilotage. Par conséquent, il nous semble qu'une architecture de pilotage hybride (proposée et explicitée dans la troisième partie de ce mémoire) engendrait une performance plus élevée en termes de réactivité et de flexibilité, et pourrait être donc prometteuse dans ce travail pour le développement d'un système de production reconfigurable et agile (SPRA).

5. Conclusion

L'objectif de ce chapitre était de présenter généralement les différents niveaux de pilotage d'un système de production reconfigurable. Après avoir défini les 5 niveaux de décision de la pyramide du CIM dans l'entreprise qui sont « ERP », « MES », « Supervision », « IHM » et « Terrain », nous avons présenté une autre approche composée de 4 niveaux de décision dans l'entreprise qui sont respectivement à partir du niveau le plus bas « Opération », « Commande », « Exécution » et « Planification ».

Ensuite, nous avons analysé les différentes architectures des systèmes de pilotage les plus souvent rencontrées dans l'industrie. Ces architectures ont été référencées classiquement en fonction de leur topologie (centralisée, hiérarchisée, distribuée, décentralisée, hybride). De plus, nous avons formé un tableau récapitulatif des caractéristiques, des avantages et des inconvénients pour ces différentes architectures afin d'évaluer l'intérêt de chacune et de proposer ensuite une architecture de pilotage convenable et prometteuse pour le développement d'un système de production reconfigurable et agile (SPRA).

Par ce chapitre, nous terminons ainsi la première partie de ce mémoire qui rend compte de l'état de l'art sur les systèmes de production et particulièrement des systèmes de production reconfigurables et agiles (**Chalfoun et al., 2011**), les outils de modélisation d'un système et le pilotage d'un système de production reconfigurable.

La partie suivante se concentre sur la phase de caractérisation et modélisation d'un système de production reconfigurable et agile (SPRA).

Deuxième Partie

Conception d'un Système de Production Reconfigurable et Agile (SPRA)

Une première étape consiste à modéliser et caractériser le système étudié qui s'agit d'un Système de Production Reconfigurable et Agile (SPRA).

SPRA est un concept innovant de système de production qui combine la reconfigurabilité et l'agilité en réagissant efficacement aux exigences imposées du marché, des clients, de la technologie des procédés, de l'environnement et de la société afin que l'entreprise soit dynamique, compétitive et rentable.

Cette partie est composée de deux chapitres, un modèle générique adapté à ce type de système sera proposé dans le premier chapitre. Ensuite, le deuxième chapitre caractérise le système étudié en focalisant sur l'aspect de la reconfiguration.

Sommaire

Chapitre 1 : Proposition d'un modèle générique d'un SPRA.....	- 109 -
Chapitre 2 : Caractérisation d'un SPRA.....	- 131 -

Chapitre 1

Proposition d'un modèle générique d'un SPRA

L'objectif de ce chapitre est de proposer un modèle générique adapté au SPRA. Ce modèle repose sur une description détaillée du SPRA selon deux axes.

Le premier axe sépare la structure du système de ses configurations, cette séparation permet de faire correspondre à une même structure un ensemble de configurations différentes. Le second axe distingue la partie logique et la partie physique, cette séparation apparaît à la fois au niveau de la structure et de la configuration. Les opérations occupent un rôle central dans le modèle puisqu'elles relient la partie physique à la partie logique.

Sommaire

1. Introduction	111
2. L'approche du Lab-STICC.....	112
3. Principe d'organisation du modèle générique.....	113
4. La structure	115
4.1. Formalisation de la structure	115
4.2. Méta-modèle haut niveau de la structure.....	116
4.3. Représentation de la structure physique	118
4.4. Représentation de la structure logique.....	120
4.5. Description des connexions	122
5. La configuration	125
5.1. Formalisation de la configuration.....	125
5.2. Méta-modèle haut niveau de la configuration	126
6. Les opérations.....	127
7. Conclusion	129

1. Introduction

Généralement, un système de production est composé d'un ensemble d'opérateurs et de postes de travail et/ou de machines qui sont intégrés pour effectuer une série contrôlée d'opérations répétitives sur les matières premières afin d'obtenir une forme finale souhaitée ou d'assembler un produit final répondant aux besoins du marché. Les postes de travail et/ou les machines sont connectés par l'intermédiaire d'un mécanisme de transfert qui peut être un convoyeur et/ou un robot mobile AGV (Automated Guided Vehicle) ou par des opérateurs humains ou une combinaison plus ou moins complexe de ces trois modes de transport.

Notre travail de recherche s'intéresse à un nouveau type de système de production SPRA qui est caractérisé par la reconfigurabilité et l'agilité afin de réagir efficacement aux différents défis, générant des changements et des situations imprévisibles imposés dans l'environnement de production (**Cunha et al., 2007**), (**Koren, 2010a**). Etant donné que le système étudié SPRA représente un système complexe d'ingénierie alors le langage SysML nous semble être le langage le mieux adapté pour modéliser ce système.

Dans ce chapitre, nous nous focalisons sur l'organisation générale d'un modèle générique adapté à ce type de système SPRA avec une description détaillée des différents éléments constituant la structure, les configurations et les opérations. Ce modèle générique repose sur une description détaillée du SPRA selon deux axes. Le premier axe sépare la structure du système de ses configurations, cette séparation permet de faire correspondre à une même structure un ensemble de configurations différentes. Le second axe distingue la partie logique et la partie physique, cette séparation apparaît à la fois au niveau de la structure et de la configuration. Les opérations occupent un rôle central dans le modèle puisqu'elles relient la partie physique à la partie logique.

Il est très important de noter que le principe d'organisation de ce modèle générique est relativement proche de celui proposé par M. Lamotte dans le cadre de sa thèse soutenue en 2006 au sein du Laboratoire des Sciences et Techniques de l'Information, de la Communication et de la Connaissance (Lab-STICC, UMR CNRS 6285) de l'Université de Bretagne Sud, ex LESTER (Laboratoire d'Electronique des Systèmes TEmps Réel). Néanmoins, la formalisation et la représentation raffinées de chaque élément de ce modèle sont complètement différentes. De plus, le principe de reconfiguration, dans la troisième partie, est totalement innovant.

2. L'approche du Lab-STICC

L'ingénierie dirigée par les modèles introduit une hiérarchisation des modèles manipulés. Un système est représenté par un modèle. De plus, un méta-modèle est un modèle définissant un modèle du niveau inférieur.

Dans le cadre des travaux de M. Pascal BERRUET et de son équipe, l'ingénierie dirigée par les modèles est considérée comme un outil majeur pour l'étude et l'implémentation des systèmes. Ainsi, le Lab-STICC a introduit un langage de représentation de haut niveau, appelé DeSyRe pour Description de Systèmes Reconfigurables (**Lamotte, 2006**).

Le principe de modélisation proposé par M. Lamotte consiste à diviser la description des systèmes de production reconfigurables selon deux axes. Le premier axe, horizontal, sépare l'architecture du système de ses configurations. Le second axe vertical de décomposition du système est l'axe logique/physique. Cet axe sépare les aspects « produit » constituant la partie logique du système, des aspects « machine » et « transport » appartenant à la partie physique. Cette séparation apparaît à la fois au niveau de l'architecture et de la configuration. En outre, les opérations sont au centre de la décomposition du système et elles relient la partie physique décrivant les ressources à la partie logique définissant les fonctions. Le lien entre l'architecture et la configuration est aussi réalisé au travers des opérations définies dans l'architecture et sélectionnées dans la configuration (Figure 2.1.1).

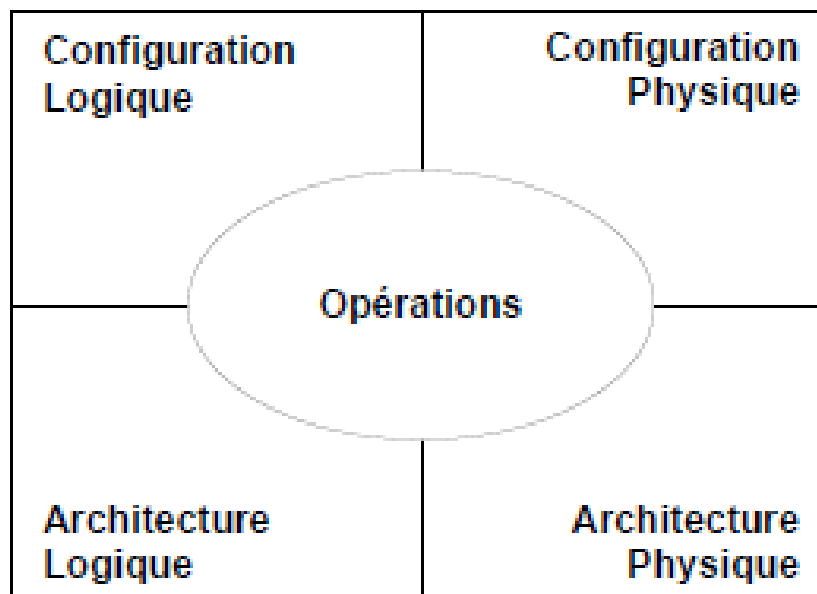


Figure 2.1.1. Organisation précédente du modèle (**Lamotte, 2006**).

Après avoir présenté en bref l'approche du Lab-STICC concernant l'organisation générale du modèle d'un système de production reconfigurable, nous présentons

dans les sections suivantes nos apports en proposant un modèle générique adapté au système étudié SPRA, et également les formalisations et les représentations raffinées de chaque élément de ce modèle.

3. Principe d'organisation du modèle générique

Dans ce paragraphe, nous expliquons comment nous organisons un modèle générique pour notre système SPRA. Nous préférons remplacer le terme architecture par le terme structure (**Chalfoun et al., 2013**).

Dans cette étude, le fonctionnement de la reconfiguration démarre toujours du besoin de lancer un nouveau produit. Donc, il faut :

- Définir les fonctions qui doivent être réalisées selon la gamme logique : Par exemple dans une usine d'assemblage mécanique, il s'agira de collage, pliage, perçage, contrôle, pointage, soudage, vissage, etc.
- Définir les composants matériels capables de réaliser ces fonctions.
- Mettre en place une plateforme de réalisation. Nous choisissons alors un ensemble de composants et nous les agençons.

La configuration à trouver est ainsi constituée de:

- ✓ Un ensemble de composants : postes de travail, convoyeurs, robots, logiciels de contrôle, etc.
- ✓ Un agencement de ces composants entre eux : par exemple une structure linéaire, parallèle ou une combinaison de ces deux.

De plus, nous parlons de deux types de configuration :

✚ Une configuration physique (matérielle) de chaque poste : nous branchons par exemple une visseuse dans un poste prévu à cet effet.

✚ Une configuration logique (logicielle) : affectation d'un programme de contrôle.

La phase de conception proposée repose sur une description du SPRA selon deux axes (Figure 2.1.2). Le premier axe, vertical, sépare la structure du système de ses configurations sachant que nous pouvons faire correspondre à une même structure un ensemble de configurations différentes. La structure définit les composants constituant le système ainsi que les connexions entre eux, la structure consiste donc en la description des ressources et leurs capacités, des produits et des moyens de communications entre ces éléments. La configuration décrit comment les différents éléments de la structure sont utilisés et organisés pour atteindre l'objectif de cette configuration. Autrement dit, la configuration représente la partie du système qui évolue selon les aléas, les modifications de l'environnement ou les modifications des objectifs fixés pour le système.

Le second axe, horizontal, distingue la partie logique et la partie physique. La structure logique est constituée des produits et des gammes logiques permettant d'obtenir ces produits. Cette partie décrit donc les produits et définit ainsi la séquence de fonctions nécessaires qui conduisent à obtenir le produit final. Au niveau de la structure physique, elle décrit l'agencement des différents composants matériels du système, les ressources stationnaires (postes de travail, tampons, ...), les ressources de transport (robots, convoyeurs, ...) ainsi que les différentes connexions entre eux. Par exemple, devraient-ils être disposés dans une ligne série, en parallèle, ou une combinaison de ces deux dispositions ? Les connexions représentent les liens de transfert potentiels entre les ressources stationnaires, ces connexions sont associées aux ressources de transport qui peuvent les exécuter.

L'aspect logique de la configuration (logiciel) correspond à l'implantation des programmes de contrôle sur chaque composant de la structure. En outre, la configuration physique (matérielle) représente les ressources utilisées et les capacités demandées de chaque ressource.

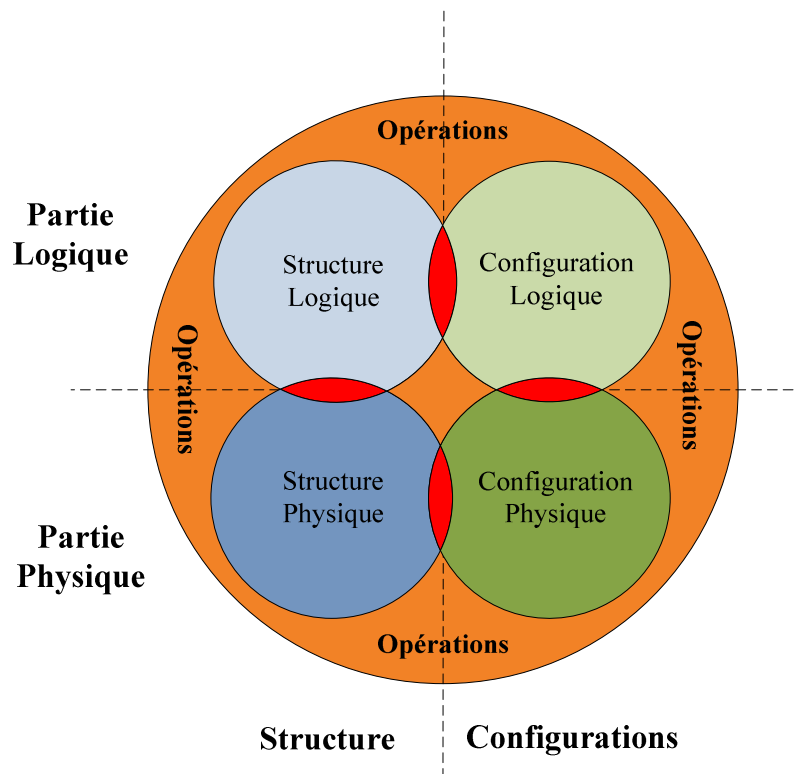


Figure 2.1.2. Principe d'organisation du modèle générique d'un SPRA.

Les opérations occupent alors un rôle central dans le modèle puisqu'elles associent les fonctions aux ressources, elles relient ainsi la structure avec les configurations et la partie physique à la partie logique. Six types d'opérations peuvent être distingués : les opérations principales qui associent les fonctions de travail aux postes de travail, les opérations de transport qui réalisent les transferts (connexions), les opérations de stockage qui assurent le stockage des produits sur les

tampons, les opérations d'affectation des ressources utilisées dans une configuration, les opérations d'activation des modules (outils et/ou dispositifs) sur les ressources utilisées selon les exigences de la configuration, et les opérations d'implantation des programmes de contrôle sur chaque ressource utilisée dans cette configuration (**Lamotte et al., 2005**), (**Lamotte, 2006**), (**Berruet, 2007**).

Pour bien organiser ce modèle du SPRA, nous allons placer la structure dans un package « Structure ». Ce package est décomposé en deux autres packages « Structure Physique » et « Structure Logique » qui contiennent respectivement les représentations de la structure physique et de la structure logique. Chaque configuration et opération sont ensuite décrites de la même façon que la structure.

Par conséquent, cette section a défini le principe d'organisation d'un modèle générique. Chaque élément de ce modèle sera formalisé et représenté dans les sections suivantes.

4. La structure

Nous avons vu que la structure définit les ressources et leurs capacités, les produits et les moyens de communication entre ces éléments.

4.1. Formalisation de la structure

La structure d'un SPRA peut être donc représentée par la paire suivante S (**Chalfoun et al., 2012b**) :

$$S: \{S_p, S_l\} \quad (2)$$

Où : S_p désigne la structure physique et S_l la structure logique.

S_p est la structure physique, décrite par le triplet :

$$S_p: \{R, Capab, Conn\} \quad (3)$$

Où : R représente l'ensemble des ressources de la structure (ressources stationnaires, ressources de transport et peut être des robots mobiles AGV), $Capab$ décrit les capacités de chaque ressource et $Conn$ représente l'application connexion tel que :

$$Conn: R_s \times R_s \times R_t \rightarrow \{0,1\} \quad (4)$$

Où : R_s représente une ressource stationnaire et R_t représente une ressource de transport.

Chaque connexion relie deux ressources stationnaires entre elles par l'intermédiaire d'une ressource de transport qui réalise le transfert.

S_l est la structure logique représentée par le quintuplet :

$$S_l: \{F, G, Pr, Prec, Affec\} \quad (5)$$

Où : F est l'ensemble des fonctions (F_{tr} : fonctions de travail, F_{ts} : fonctions de transport et F_{st} : fonctions de stockage), G est l'ensemble des gammes logiques (séquences de fonctions) et Pr est l'ensemble des types de produits.

L'application $Prec$ définit la relation de précédence entre les fonctions au sein d'une gamme logique :

$$Prec: G \times F \times F \rightarrow \{0,1\} \quad (6)$$

L'application $Affec$ permet d'affecter une ou plusieurs gammes logiques à tout type de produits :

$$Affec: G \times Pr \rightarrow \{0,1\} \quad (7)$$

4.2. Méta-modèle haut niveau de la structure

Le langage de modélisation SysML offre le diagramme de définition de bloc (BDD : Block Definition Diagram) qui est basé sur le diagramme de classes UML. Ce diagramme nous permet de représenter de manière générique la structure en unités modulaires « blocs » avec les relations entre eux (Figure 2.1.3).

Cette figure représente la structure S d'un point de vue particulier ce que nous appelons « méta-modèle » qui montre les composants modulaires à construire, d'une façon « plus simple » et « plus claire », et le réseau de relations qui assure le flux de données entre ces composants. L'objectif de ce méta-modèle est de construire la structure du SPRA d'une façon modulaire qui va faciliter la reconfiguration du système et collaborer avec les autres composants pour réaliser l'agilité souhaitée.

Le bloc « Structure » représente la structure du système et possède deux relations, avec la structure physique ainsi qu'avec la structure logique.

La structure physique est composée des ressources et des connexions. Chaque ressource possède des capacités, trois types de ressources sont considérés : ressources stationnaires (postes de travail, postes de secours et tampons), ressources de transport (convoyeurs, robots et chargeur/déchargeur) et des robots mobiles AGV si besoin. Pour ce qui est des postes de travail, le bloc « Ports » définit les lieux caractéristiques pour les transferts. Les connexions sont représentées par le bloc « Connexions », elles relient les ports de postes de travail par l'intermédiaire d'un convoyeur ou d'un robot.

Au niveau de la structure logique, trois types de fonctions sont définies : fonctions de travail qui interviennent dans la réalisation de produits prenant en compte aussi les fonctions de contrôle-inspection, fonctions de transport qui sont associées aux connexions et fonctions de stockage qui relient le chargeur/déchargeur aux tampons. Le bloc « Produits » représente les types de produits décrits par les gammes logiques « Gamme » qui représentent les séquences de fonctions. Le bloc « Précédence » définit la relation de précédence entre les fonctions au sein d'une

gamme logique. Le bloc « Affectation » associe une ou plusieurs gammes logiques à tout type de produits (Chalfoun *et al.*, 2013).

Nous avons exprimé tout cela sur la Figure 2.1.3 en utilisant Artisan Studio par les trois types de relations standardisées qui sont (Holt *et al.*, 2008), (Atego, 2010), (Friedenthal *et al.*, 2012):

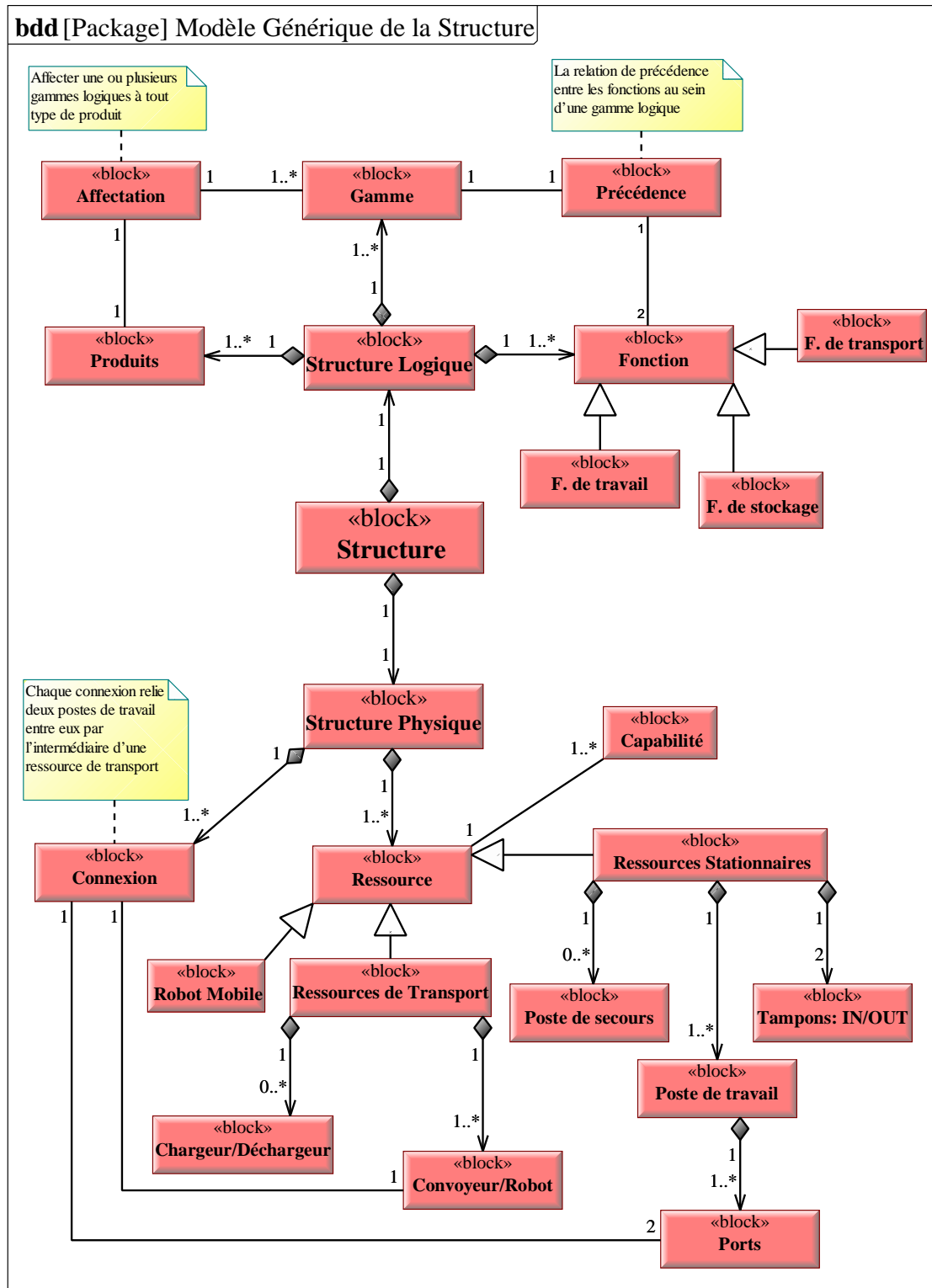


Figure 2.1.3. Méta-modèle haut niveau de la structure.

- « **Association** » : il s'agit d'une spécification des liens que deux blocs peuvent établir.
- « **Part association** » : c'est une relation de type composant/composé forte, le cycle de vie du composant est lié à celui du composé.
- « **Generalization** » : un mécanisme par lequel on « factorise » des caractéristiques communes à plusieurs entités dans une entité générique.

Effectivement, avec un bloc, nous pouvons définir tout ce qu'un utilisateur SysML peut juger utile à ses besoins, par exemple : des contraintes ou des références vers des contraintes, des fonctions, des caractéristiques structurelles (propriétés), des compositions en « sous-bloc » (parts), des associations avec d'autres blocs (références), des caractéristiques quantifiables et dimensionnées, des ports, des unités et dimensions, etc.

Donc, chaque bloc est défini par les caractéristiques qu'il possède, qui peuvent être divisées en caractéristiques structurelles et caractéristiques comportementales (Figure 2.1.4).

« Block » Nom du bloc
Contraintes
Opérations
Parts
Ports
Dimensions et unités

Figure 2.1.4. Caractéristiques d'un bloc.

4.3. Représentation de la structure physique

Cette partie de la structure décrit l'organisation des différents éléments matériels constituant le système tels que les ressources stationnaires (postes de travail, tampons, ...), les ressources de transport (robots, convoyeurs, ...) ainsi que les différentes connexions entre elles.

La structure physique du système, sur laquelle seront réalisées les fonctions de travail demandées, peut être décrite par le schéma suivant en termes de ports et de connexions (Figure 2.1.5).

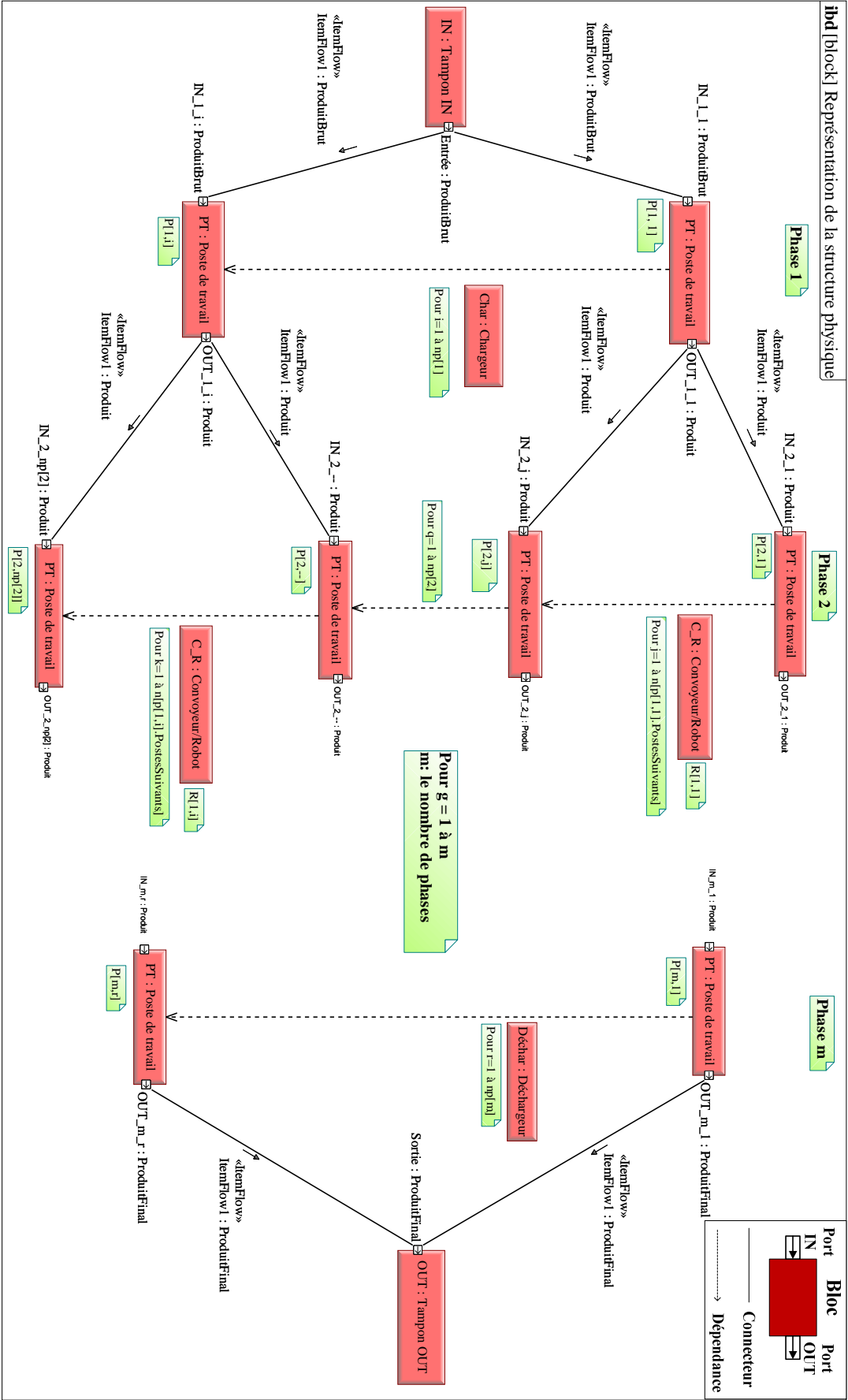


Figure 2.1.5. Représentation de la structure physique.

Les ressources stationnaires (postes de travail, tampons IN/OUT) et les ressources de transport (convoyeur/robot, chargeur/déchargeur) sont représentées par des blocs. Les ports associés aux postes de travail sont représentés à l'aide d'une flèche IN ou OUT. Les connexions sont décrites par un connecteur dirigé du port de sortie d'un bloc vers le port d'entrée du bloc suivant. Les ressources de transport sont reliées aux connexions, et bien sûr selon le nombre de postes suivants du poste actuel, par l'intermédiaire d'une dépendance (représentée par un trait discontinu dont l'extrémité porte une flèche).

4.4. Représentation de la structure logique

La structure logique est constituée des produits et des séquences de fonctions permettant d'obtenir ces produits en réalisant des fonctions. Cette partie décrit donc les fonctions de travail à réaliser sur le produit et définit ainsi la gamme logique (séquences de fonctions nécessaires) qui conduit à obtenir le produit final.

Dans notre recherche, nous voulons utiliser la notion de « produit intelligent » qui est muni d'un système d'auto-identification (Auto-ID) à l'aide, par exemple, d'une étiquette RFID qui permet de relier l'identifiant du produit à la commande du système de production (McFarlane *et al.*, 2003), (Meyer *et al.*, 2009).

Le produit est donc porteur d'informations, son identification va permettre de conduire le système en informant chaque poste de travail de ce qu'il faut faire. De même, ce produit peut être transformé et/ou transporté entre les composants du système.

La structure logique est constituée de j produits (Pr_1, \dots, Pr_j) décrits par les gammes logiques (G_1, \dots, G_j) qui sont définies comme des séquences de fonctions de travail ($F_{tr_1}, \dots, F_{tr_n}$), des fonctions de transport ($F_{ts_1}, \dots, F_{ts_n}$) qui sont associées aux ressources de transport et des fonctions de stockage ($F_{st_{in}}, F_{st_{out}}$) qui sont associées aux tampons IN et OUT respectivement.

Le diagramme suivant représente la structure logique du SPRA en définissant uniquement ses éléments principaux (Figure 2.1.6). Les produits, le produit brut et le produit final, sont représentés par des blocs « Produits », « Produit Brut » et « Produit Final » respectivement. Les gammes logiques sont affichées comme des blocs « Gamme ». Les fonctions sont définies sous la forme d'un bloc « Fonction de Travail » ou « Fonction de Transport » ou « Fonction de Stockage ».

Au niveau des gammes logiques, celles-ci peuvent être ensuite décrites par des diagrammes d'activité qui sont composés d'un ensemble d'activités chaînées. Ce type de diagramme permet de modéliser des algorithmes complexes dans le cadre de la modélisation dynamique. Il est utilisé majoritairement pour l'expression de la logique de contrôle et d'entrées/sorties. Le diagramme d'activité représente ainsi les flots de données et de contrôle entre les actions. Une action représente un traitement

ou une transformation, les actions sont contenues dans les activités. Un flot est un contrôle de séquençage pendant l'exécution des activités.

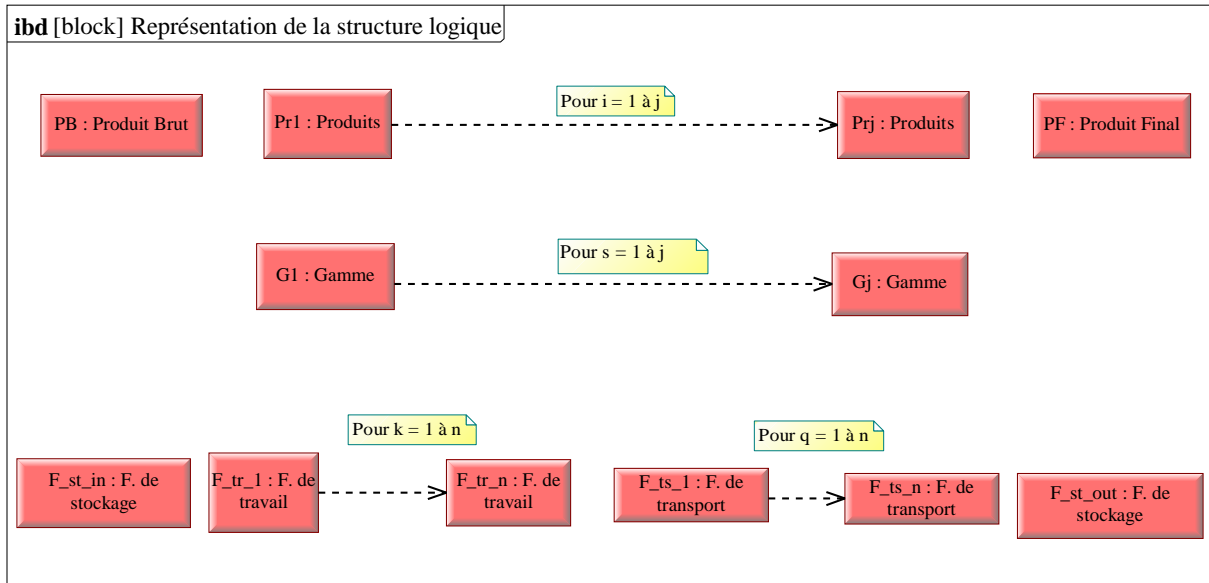


Figure 2.1.6. Représentation de la structure logique.

Les diagrammes d'activités sont des diagrammes comportementaux SysML très importants et efficaces, qui peuvent être utilisés pour montrer à la fois le niveau inférieur du comportement, telles que les opérations, ainsi que le haut niveau du comportement, tels que les flux de travail. Ils aident aussi à assurer la cohérence du modèle concernant les diagrammes d'état, les diagrammes de séquence et les diagrammes de définition de blocs (**Holt et al., 2008**). Nous allons ainsi construire en détails quelques diagrammes d'activité selon le besoin dans la troisième partie de ce mémoire « Déploiement d'un SPRA ».

Dans le diagramme suivant (Figure 2.1.7), nous présentons un exemple général des gammes logiques. Nous considérons que nous voulons obtenir un produit final Pr_j à partir du produit brut Pr_0 . Soit G_j la gamme logique associée à la fabrication de ce produit alors cette gamme est définie comme des séquences de fonctions nécessaires, soit $G_j = (F_{st_in}, F_{ts_1}, F_{tr_1}, \dots, F_{st_out})$.

La Figure 2.1.7 illustre les produits, les activités et les flots de données entre elles. Les produits sont représentés par le nœud initial et le nœud final, chaque activité est reliée à une fonction. Le produit en entrée Pr_0 est relié à la première activité de la séquence et le produit en sortie Pr_j à la dernière. Les flots de contrôle sont de simples flèches reliant deux activités (**Atego, 2010**).

L'activité démarre avec la première fonction F_{st_in} . Lorsque la fonction F_{st_in} est terminée, elle produit un jeton qui va alimenter la fonction suivante F_{ts_1} , via le flot de contrôle. En réalité, le produit intelligent va alimenter la fonction suivante sur un autre poste de travail, cette fonction démarre alors et possède sa propre durée.

Lorsque la dernière fonction F_st_out est activée, elle s'exécute et son exécution fait également terminer cette activité.

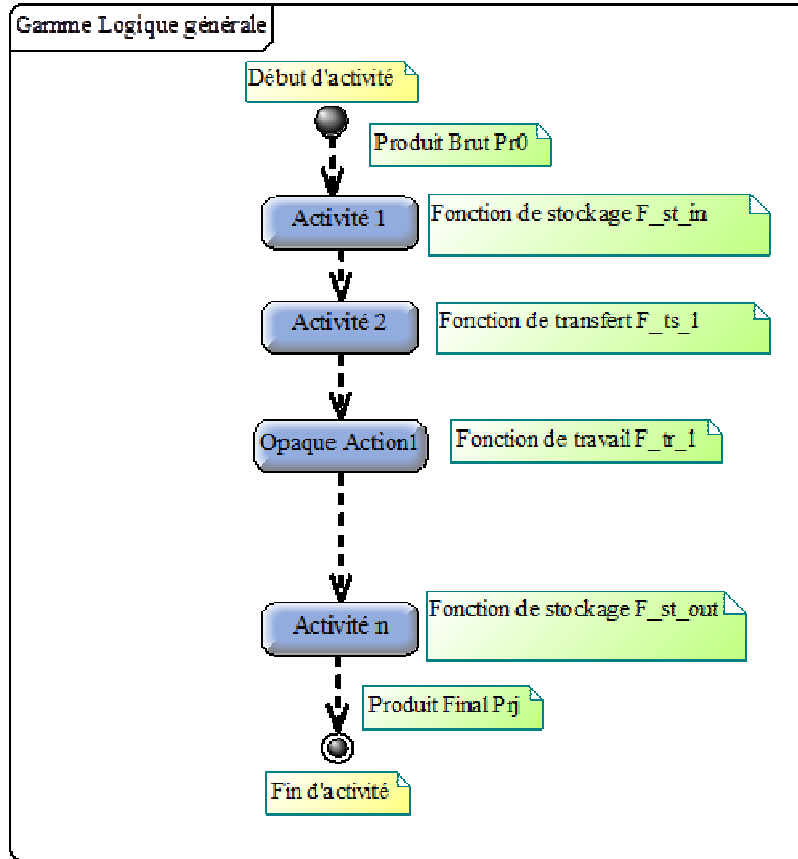


Figure 2.1.7. Représentation générale d'une activité d'une gamme logique.

4.5. Description des connexions

Les connexions représentent les liens de transfert potentiels entre les postes de travail. Ces connexions sont associées aux ressources de transport (convoyeurs, robots) qui peuvent les exécuter.

Afin de décrire les connexions, nous supposons qu'une configuration soit déjà donnée. Cette configuration définit donc : le nombre de postes de travail N et les deux tampons IN et OUT, le nombre de phases m avec les deux phases de IN et OUT, le nombre de postes à chaque phase $np[m]$, l'arrangement de postes, les connexions entre eux, les coordonnées cartésiennes (x_i, y_i) pour chaque poste au niveau de l'atelier et (x_{in}, y_{in}) , (x_{out}, y_{out}) les coordonnées de IN et OUT respectivement.

Nous représentons le poste de travail de la phase i et de numéro local j par $P[i, j]$ comme un élément de la matrice transposée de :

$$P = (p_{i,j})_{1 \leq i \leq m, 1 \leq j \leq \max(np[m])} \quad (8)$$

Où : $\max(np[m])$ est le maximum entre les nombres de poste à chaque phase.

La ressource de transport associée à $P[i, j]$ est $R[i, j]$, les coordonnées cartésiennes du poste $P[i, j]$ sont données par :

$$P[i, j].coordonnées = (x_i, y_i) \quad (9)$$

Nous rappelons que notre objectif est d'intégrer la notion d'agilité avec la fonction de reconfiguration aux systèmes de production. Pour cela, nous proposons de définir les postes suivants à chaque poste de travail à partir de IN à OUT considérant que cela participera et aidera l'agilité du système.

Les postes suivants de $P[i, j]$ peuvent être représentés par un vecteur :

$$P[i, j].PostesSuivants$$

Le nombre de postes suivants de $P[i, j]$ est donné par :

$$\text{length}[P[i, j].PostesSuivants]$$

Nous définissons les connexions dans l'atelier considérant que X_Y signifie que X est en connexion avec Y :

Entre IN et la première phase (cas: $m=1$), les connexions sont réalisées à l'aide du chargeur :

Pour $i=1$ à $np[1]$

$IN_P[1, i]$ réalisée par le chargeur ;

Entre la dernière phase m et OUT, les connexions sont réalisées à l'aide du déchargeur :

Pour $i=1$ à $np[m]$

$P[m, i]_{OUT}$ réalisée par le déchargeur ;

Entre la phase 1 et la phase m , les connexions sont réalisées à l'aide de la ressource de transport (convoyeur, robot) associée au poste $P[i, j]$:

Pour $i=1$ à $m-1$

Pour $j=1$ à $np[i]$

Pour $k=1$ à $\text{length}[P[i, j].PostesSuivants]$

$P[i, j]_{P[i, j].PostesSuivants[k]}$ réalisée par le $R[i, j]$;

Nous présentons un exemple (Figure 2.1.8) afin que l'idée soit plus claire, nous supposons donc que nous avons besoin de réaliser un produit Pr décrit par la gamme logique G .

Soient $N=10$ le nombre de postes de travail nécessaire, $m=3$ le nombre de phases, le nombre de postes à chaque phase : $np[1] = 2$, $np[2] = 4$, $np[3] = 4$. Nous rappelons que les coordonnées sont : $IN.coordonnées = (x_{in}, y_{in})$, $OUT.coordonnées = (x_{out}, y_{out})$, et $P[i, j].coordonnées = (x_i, y_i)$ les coordonnées des postes $P[i, j]$.

Les postes suivants à chaque poste sont définis comme la suite :

$IN.PostesSuivants = [P[1, 1], P[1, 2]]$
 $P[1, 1].PostesSuivants = [P[2, 1], P[2, 2]]$
 $P[1, 2].PostesSuivants = [P[2, 3], P[2, 4]]$
 $P[2, 1].PostesSuivants = P[3, 1]$
 $P[2, 2].PostesSuivants = P[3, 2]$
 $P[2, 3].PostesSuivants = P[3, 3]$
 $P[2, 4].PostesSuivants = P[3, 4]$
 $[P[3, 1], P[3, 2], P[3, 3], P[3, 4]].PostesSuivants = OUT$

La Figure 2.1.8 représente toutes les connexions entre les postes de travail, et également la ressource de transport associée à chacun au niveau de l'atelier.

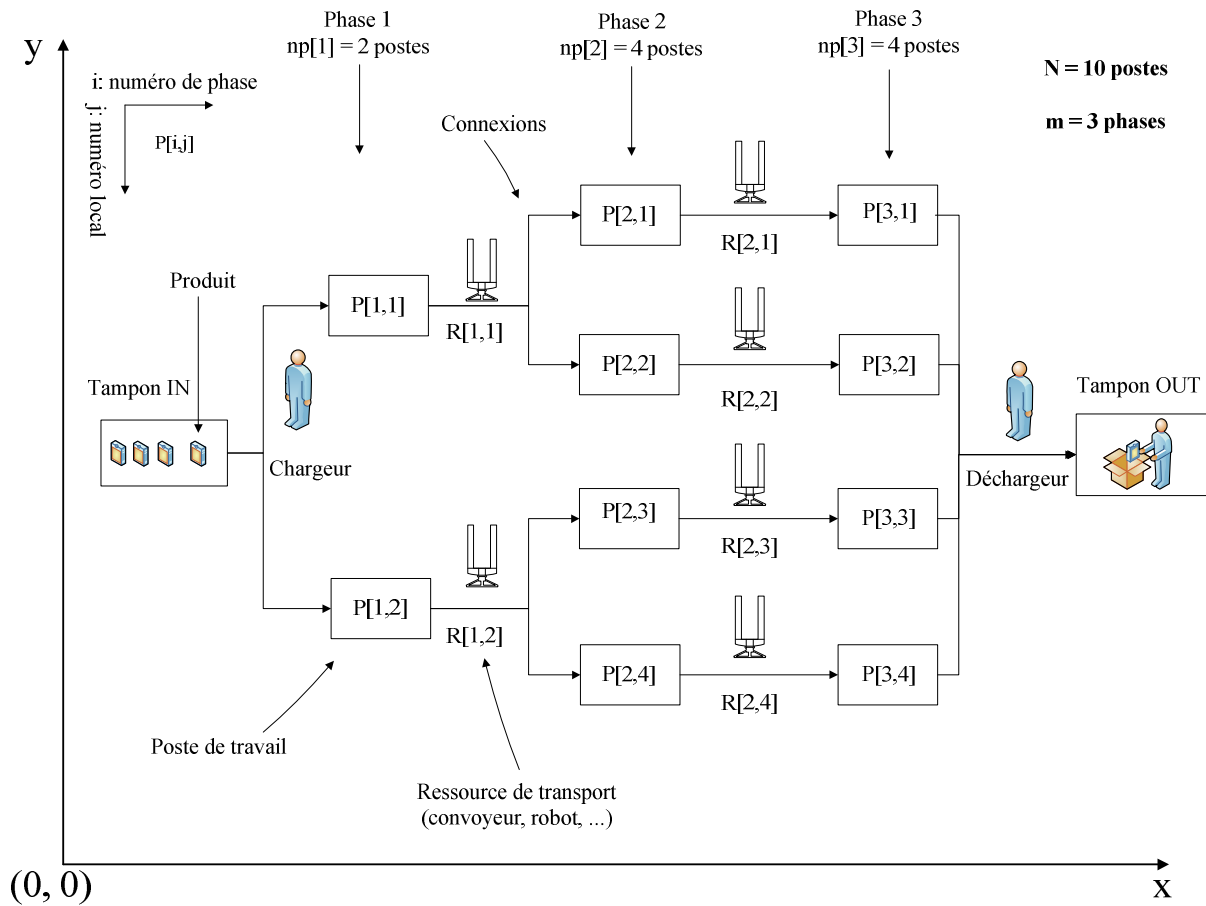


Figure 2.1.8. Exemple d'une structure avec les connexions.

En conséquence, nous avons représenté les connexions de telle manière à déplacer et connecter facilement les postes de travail à partir de ses coordonnées et

ses voisins. En effet, cela traduit la mobilité du système qui permet de reconfigurer rapidement le système, et ainsi de contribuer à l'agilité souhaitée du SPRA.

5. La configuration

Le modèle générique proposé pour SPRA a séparé la structure du système de ses configurations en partant de l'hypothèse que nous pouvons relier la même structure à un ensemble de configurations différentes. Nous avons détaillé la structure d'un SPRA tel que nous avons vu sa formalisation proposée, sa description et ses différentes représentations. D'autre part, les configurations définissent les différentes utilisations et organisations des éléments de la structure pour que chaque configuration atteigne son objectif.

5.1. Formalisation de la configuration

Nous pouvons représenter chaque configuration par la paire suivante C (Chalfoun *et al.*, 2012b) :

$$C: \{C_p, C_l\} \quad (10)$$

Où: C_p est la configuration physique et C_l la configuration logique.

C_p définit la configuration physique, décrite par la paire :

$$C_p: \{R_{util}, Capab_{assoc}\} \quad (11)$$

Où: R_{util} représente l'ensemble des ressources utilisées dans cette configuration, cet ensemble est inclus dans l'ensemble global des ressources R . De plus, $Capab_{assoc}$ définit les capacités demandées et associées à chaque ressource et ainsi, pour chaque ressource, quelques modules (outils et/ou dispositifs) pourront être utilisés.

En outre, l'application $RessUtil$ permet de déclarer les ressources utilisées parmi les ressources de la structure dans cette configuration :

$$RessUtil: R_{util} \times R \rightarrow \{0,1\} \quad (12)$$

Au niveau de chaque ressource, l'application $CapabAssoc$ permet de préciser les capacités associées pour cette configuration :

$$CapabAssoc: Capab_{assoc} \times Capab \rightarrow \{0,1\} \quad (13)$$

Donc, la configuration physique est un indicateur principal pour évaluer cette configuration au niveau du coût.

Par contre, la configuration logique représente l'ensemble des programmes de contrôle correspondant aux produits, aux gammes logiques ainsi qu'aux configurations :

$$C_l: \{Prog\} \quad (14)$$

L'application *Implant* permet ainsi d'implanter un programme de contrôle sur chaque ressource utilisé dans la configuration actuelle :

$$\text{Implant: Prog} \times R_{\text{util}} \rightarrow \{0,1\} \quad (15)$$

5.2. Méta-modèle haut niveau de la configuration

Nous allons représenter la configuration en composants modulaires « blocs » comme nous avons fait pour la structure. Le diagramme de définition de blocs (Figure 2.1.9) illustre les méta-modèles haut niveau d'une configuration en essayant de réaliser la fonction de reconfiguration et d'agilité d'un SPRA d'une façon claire et pratique en utilisant Artisan Studio (Holt et al., 2008), (Friedenthal et al., 2012).

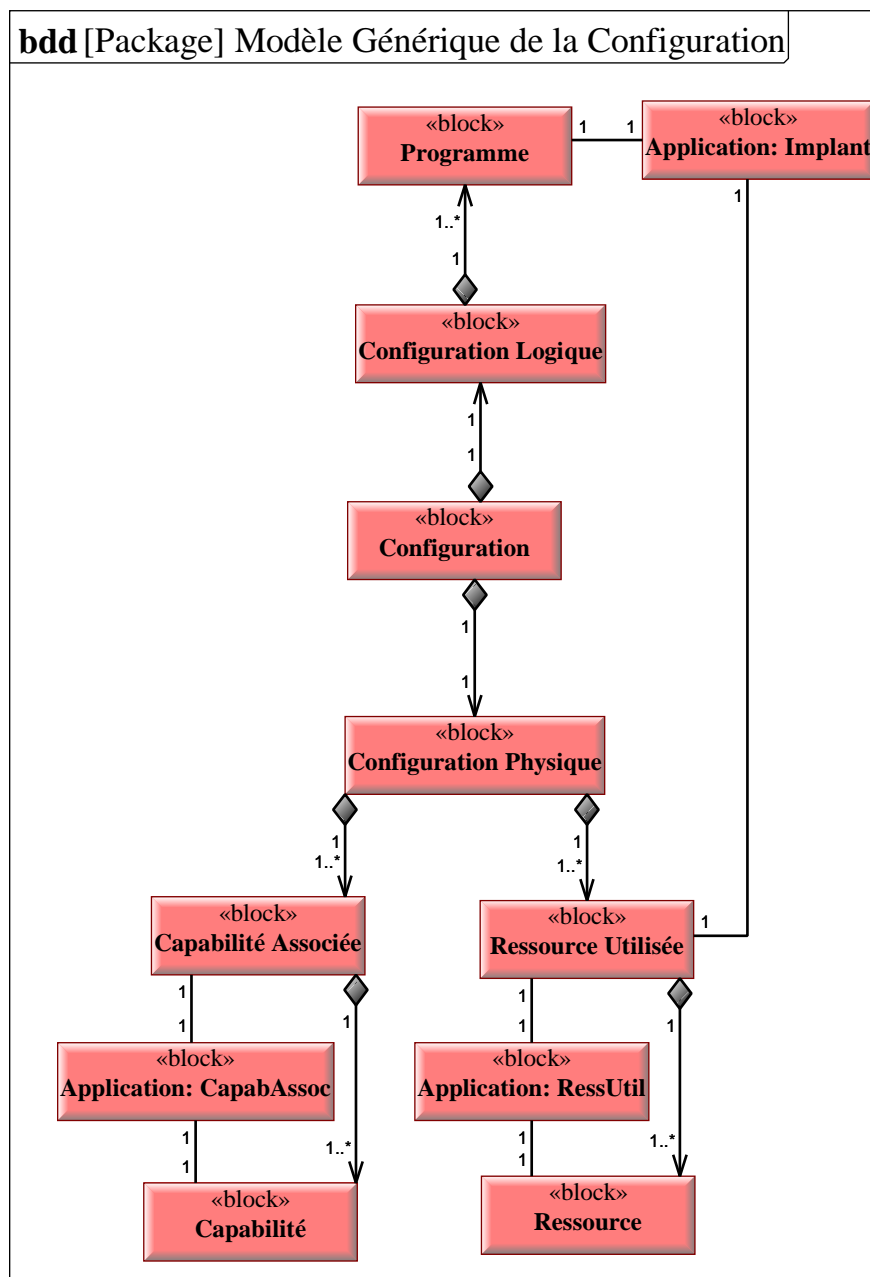


Figure 2.1.9. Méta-modèle haut niveau de la configuration.

Le bloc « Configuration » représente une configuration possible du système et possède deux relations avec la configuration physique et la configuration logique.

D'une part, la configuration physique est constituée de l'ensemble des ressources utilisées et des capacités demandées et associées à chaque ressource pour cette configuration. D'autre part, la configuration logique représente l'ensemble des programmes de contrôle correspondant aux produits, aux gammes logiques ainsi qu'aux configurations (**Chalfoun et al., 2013**).

Toutes les applications déjà définies sont illustrées sur ce diagramme de définition de blocs.

6. Les opérations

Dans cette section, nous décrivons les opérations qui sont un élément principal du modèle générique.

Nous avons distingué six types d'opérations au niveau de l'atelier : les opérations principales, les opérations de transports, les opérations de stockage, les opérations d'activation, les opérations d'affectation et les opérations d'implantation (**Chalfoun et al., 2013**). Ces opérations relient la structure du système avec ses différentes configurations possibles et relient aussi la partie physique à la partie logique en utilisant Artisan Studio (Figure 2.1.10).

Les opérations principales sont définies par le bloc « Op. Principales », il s'agit d'une instance d'une fonction de travail à un poste de travail associé. Chaque poste de travail doit réaliser au moins une opération principale associée à une fonction de travail, par exemple : $Op_1 (PT_1, F_{tr_1}), \dots, Op_r (PT_k, F_{tr_n})$. En plus, chaque fonction peut être implémentée au moins par une opération, par exemple, F_{tr_1} est implémentée par 2 opérations réalisées sur PT_1 et PT_3 , nous définissons alors les deux opérations $Op_1 (PT_1, F_{tr_1})$ et $Op_2 (PT_3, F_{tr_1})$. L'application suivante décrit les opérations implantant les fonctions de travail sur les postes de travail associés :

$$Op_{principale}: Op \times F_{tr} \times R_s \rightarrow \{0,1\} \quad (16)$$

Par ailleurs, les opérations de transport sont représentées dans le bloc « Opérations de transports », chaque opération de transport est reliée à un transfert (une connexion) et à une instance de fonction de transport correspondante. La deuxième application définit les opérations associant les fonctions de transport aux différentes connexions :

$$Op_{transport}: Op \times F_{ts} \times Conn \rightarrow \{0,1\} \quad (17)$$

Au niveau des opérations de stockage, elles assurent le stockage des produits sur les tampons IN et OUT, elles sont décrites dans le bloc « Op. de stockage ». $St_0 (Pr_0, IN)$ permet de stocker le produit brut Pr_0 sur le tampon IN et $St_j (Pr_j, OUT)$ permet de stocker le produit final (finalisé sur un des postes de travail de la phase m) sur le

tampon OUT. Nous définissons ainsi une application qui décrit les opérations permettant de stocker les produits sur les tampons IN et OUT :

$$Op_stockage: Op \times F_st \times Tampons \rightarrow \{0,1\} \quad (18)$$

Les opérations d'affectation permettent d'affecter les ressources utilisées dans une configuration. Cette partie de ressources sera alors éligible à utiliser :

$$Op_affectation: Op \times R_util \rightarrow \{0,1\} \quad (19)$$

En ce qui concerne les opérations d'activation, il s'agit d'activer des modules (outils et/ou dispositifs) sur les ressources utilisées selon les exigences de la configuration. Par exemple, nous branchons une visseuse et en même temps nous débranchons une perceuse dans un poste prévu à cet effet :

$$Op_activation: Op \times Capab_assoc \times R_util \rightarrow \{0,1\} \quad (20)$$

Les opérations d'implantation permettent d'implanter des programmes de contrôle sur chaque ressource utilisée dans cette configuration. Par exemple, au sein d'une gamme logique, nous activons les opérations demandées d'un poste et en même temps nous désactivons les autres opérations que ce poste est capable de faire :

$$Op_implantation: Op \times Prog \times R_util \rightarrow \{0,1\} \quad (21)$$

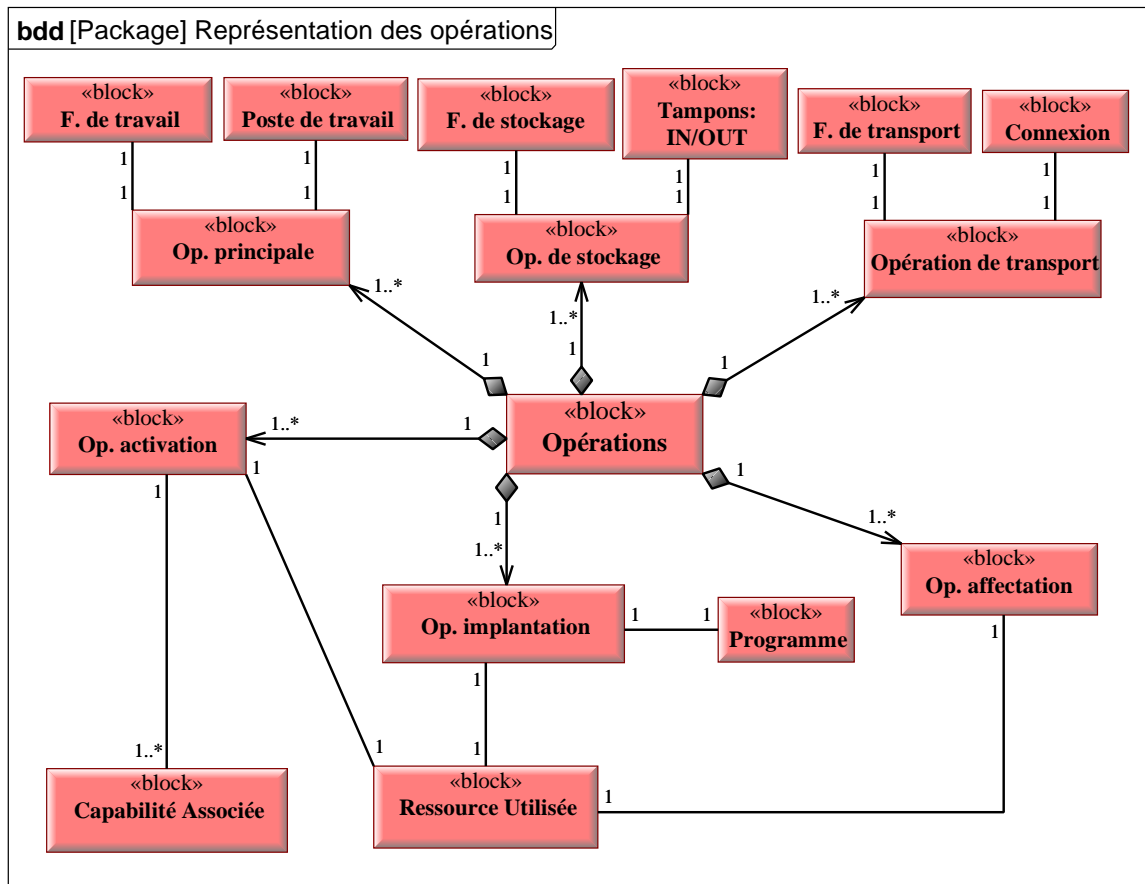


Figure 2.1.10. Représentation des opérations.

En outre, les séquences d'opérations peuvent être représentées par un bloc « Séquences d'opérations ». Ainsi chaque séquence d'opérations est associée à une gamme logique d'un produit. Une séquence d'opérations est un enchaînement des opérations permettant de réaliser la gamme logique d'un produit.

En conséquence, les opérations représentent un autre indicateur principal pour évaluer cette configuration au niveau du temps opératoire.

7. Conclusion

Le but de ce chapitre était de proposer un modèle générique adapté à ce nouveau système de production SPRA.

Nous avons présenté nos apports en proposant une approche qui repose sur une description du système selon deux axes. Le premier axe sépare la structure du système de ses configurations, le second axe distingue la partie logique et la partie physique. De plus, les opérations jouent un rôle central dans cette organisation. Chaque élément de ce modèle a été ensuite formalisé et représenté en composants modulaires « blocs ».

Dans le chapitre suivant, nous analysons les différentes exigences à plusieurs niveaux pour une application au sein du laboratoire Institut Pascal (IP). Une caractérisation du SPRA sera proposée et nous présenterons une description de haut niveau de la structure et des fonctionnalités du futur système SPRA en se focalisant sur la fonction de reconfiguration. Les acteurs et les principaux cas d'utilisation seront ensuite identifiés.

Chapitre 2

Caractérisation d'un SPRA

Le chapitre précédent a introduit un modèle générique adapté au SPRA avec une description détaillée des différents éléments constituant la structure, les configurations et les opérations.

Dans ce chapitre, nous spécifions la fonction de reconfiguration d'un SPRA afin que nous puissions la développer. Dans un premier temps, nous définissons les exigences permettant de spécifier les fonctions que le système devra réaliser, et également les contraintes qui doivent être satisfaites par ce futur système SPRA. Ensuite, nous proposons une description de haut niveau des fonctionnalités et de la structure du système.

Sommaire

1. Introduction	133
2. Etude des exigences.....	133
2.1. Typologie des exigences	133
2.2. Définition et analyse des exigences du SPRA	134
2.3. Diagrammes des exigences	136
2.4. Allocation des exigences	139
3. Description de haut niveau des fonctionnalités du SPRA.....	142
4. Description structurelle du SPRA	147
5. Conclusion.....	149

1. Introduction

Dans le cadre de cette thèse, nous travaillons pour contribuer à la naissance d'un SPRA qui est un concept innovant de système de production en combinant la reconfigurabilité et l'agilité pour le rendre dynamique et réactif contre les défis et les changements imposés dans l'environnement de production.

Nous avons présenté dans le chapitre précédent une approche d'un modèle générique adapté au SPRA avec une description détaillée de ses différents éléments.

L'objectif de ce chapitre est de caractériser les besoins de reconfiguration d'un SPRA en définissant et analysant les exigences fonctionnelles et non fonctionnelles tant au niveau système qu'au niveau de ses différents constituants, et aussi en proposant une description fonctionnelle et structurelle. Les acteurs et les principaux cas d'utilisation seront identifiés. Le langage de modélisation SysML nous permettra de représenter graphiquement les exigences et également les cas d'utilisation à plusieurs niveaux.

2. Etude des exigences

2.1. Typologie des exigences

Par définition, une exigence précise une capacité ou une condition qui doit être satisfaite par le système. Généralement, une exigence peut spécifier une fonction que le système doit réaliser ou une condition de performance ou de sécurité à atteindre. En plus, les exigences peuvent être identifiées à plusieurs niveaux.

Le processus de définition et d'analyse des exigences a pour objectif de définir un ensemble d'exigences complet et cohérent partagé avec le client. En effet, il n'existe pas de typologie standardisée associée aux exigences, mais les exigences peuvent être systématiquement formalisées sous la forme d'exigences fonctionnelles traduisant les fonctions demandées du système ou bien ce que doit faire le système, et également sous la forme d'exigences non fonctionnelles traduisant les contraintes imposées au système comme les conditions de performance, de fiabilité, de sécurité, etc. L'ensemble cohérent et complet des exigences est appelé la spécification du système et notamment lorsqu'il est validé et agréé par les parties prenantes. De plus, il est important de noter que la spécification des exigences est considérée comme un processus continu et itératif dans la mesure où les choix de conception peuvent générer de nouvelles exigences globales à prendre en compte au niveau système ou de nouvelles exigences détaillées résultant de l'allocation des exigences système sur les sous fonctions et constituants (Figure 2.2.1). (IEEE 1220, 1999), (Joergensen *et al.*, 2011), (AFIS, 2013).

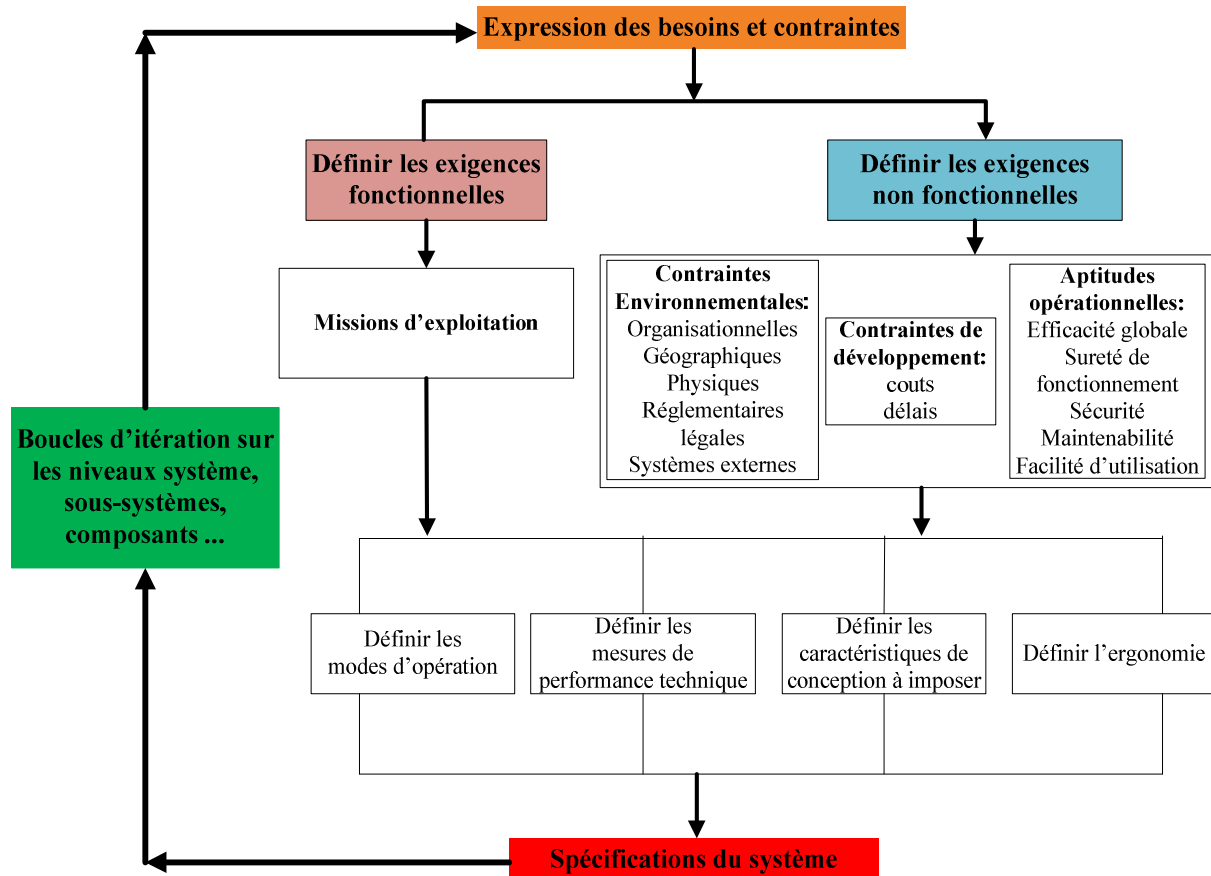


Figure 2.2.1. Typologie des exigences (Meinadier, 2002), (Petin, 2007).

2.2. Définition et analyse des exigences du SPRA

L'objectif principal de notre travail est de pouvoir reconfigurer le système rapidement pour qu'il soit adapté aux nouvelles exigences de production et aux évolutions radicales de son environnement dont certaines peuvent être totalement imprévues.

Dans ce contexte, nous avons besoin de caractériser la fonction de reconfiguration d'un SPRA afin que nous puissions la développer. Pour cela, nous définissons les exigences permettant de spécifier les capacités du système, et les contraintes qui doivent être satisfaites par ce SPRA. Autrement dit, ces exigences peuvent spécifier une fonction que le système devra réaliser en répondant aux besoins du client ou une condition de performance, de fiabilité et de sécurité que le SPRA devra respecter. Spécifier un SPRA impose donc de définir et d'analyser l'ensemble des exigences tant au niveau système qu'au niveau de ses différents constituants. Pour un SPRA (Chalfoun *et al.*, 2014), nous avons identifié l'ensemble des exigences décrites ci-dessous (Tableau 2.2.1) :

Exigences de Reconfigurabilité	
1	Le système doit permettre de passer d'une configuration à une autre.
2	Le système doit être modulable tel que tous ses composants doivent être modulaires.
3	Le système doit rendre possible la reconfigurabilité matérielle/logistique et des processus.
Exigences d'Agilité	
4	Tous les composants du système doivent être facilement déplaçables dans l'atelier.
5	Le temps de reconfiguration doit être inférieur à une certaine valeur définie par le client.
6	Toutes les ressources (postes de travail, robots, ...) doivent posséder des coordonnées cartésiennes modifiables.
Exigences Fonctionnelles	
7	Le système doit rendre possible la production des produits selon leurs gammes logiques.
8	Le système doit permettre de lancer des nouveaux produits.
9	Le système doit permettre de rechercher ou choisir une nouvelle configuration adaptée à chaque situation.
10	Le système doit permettre d'implanter et déployer une nouvelle configuration.
11	Le système doit rendre possible le pilotage et la supervision (gestion et suivi de production, de performance, de familles d'arrêts).
Exigences de Performance	
12	Le système doit être capable de satisfaire les besoins du client (quantités de produits) dans les délais impartis.
13	Le système doit respecter la qualité des produits.

Exigences d'Interfaces	
14	Le système doit rendre possible les différentes communications entre ses différents constituants.
15	Chaque ressource (postes de travail, convoyeurs, robots, ...) doit être adaptée à l'atelier (ressources de l'énergie, réseau,...).
Exigences Opérationnelles	
16	Ergonomie : toutes les ressources doivent être ergonomiques.
17	Sûreté de fonctionnement : le système doit être testable, fiable et sécurisé.
18	Environnement opérationnel : le système doit fonctionner dans les conditions normales de production (température, pression, humidité, ...).
19	Maintenance : le système et ses différents constituants doivent pouvoir être diagnostiqués et donc maintenables.
Exigence Economique	
20	Coût : le coût de reconfiguration doit être raisonnable pour le configurateur. Ainsi, le système doit rendre possible l'estimation du coût de reconfiguration : le coût nécessaire pour passer d'une configuration à une autre, et aussi le coût de production avec la nouvelle configuration qui concerne notamment la consommation d'énergies, d'équipement et d'outillage (configuration physique) afin d'avoir une idée sur l'efficacité et la rentabilité de la reconfiguration.

Tableau 2.2.1. Ensemble des exigences d'un SPRA.

2.3. Diagrammes des exigences

Le nouveau langage de modélisation SysML innove en proposant un diagramme d'exigences permettant de représenter graphiquement les exigences système dans le modèle et surtout de les relier aux éléments structurels ou dynamiques de la modélisation, ainsi qu'à d'autres exigences de niveau sous-système ou équipement. Afin de structurer la modélisation des différents types d'exigences à différents niveaux d'abstraction, le diagramme des exigences de SysML capture les hiérarchies d'exigences, donne la possibilité d'organiser les exigences de façon précise, permet de décomposer chacune d'elles et formalise les liens entre exigences sous la forme

de relations entre objets (**Holt et al., 2008**), (**Roques, 2009**), (**Atego, 2010**). En conséquence, le diagramme d'exigences sert à définir un modèle conceptuel de traçabilité et à saisir des liens effectifs dans les travaux de modélisation. Par contre, les diagrammes d'exigences ne sont pas des diagrammes d'architecture système, mais un support aux travaux d'architecture.

Au niveau des acteurs principaux dans notre système SPRA, nous identifions le configurateur, le chef de l'atelier, les opérateurs et les opérateurs de la maintenance. Ainsi, nous allons relier chacun des acteurs aux exigences correspondantes afin d'obtenir des demandes simples, et également d'être capable d'affecter et de lier ces exigences à des éléments ciblés du modèle. Par exemple, un diagramme des exigences (Figure 2.2.2), en utilisant Artisan Studio, regroupe les exigences relatives au configurateur formulant des attentes sur la reconfiguration du système, la recherche et l'implantation d'une configuration. Ces deux dernières exigences sont contenues dans la reconfiguration par la relation de contenance « Containment », et l'exigence de reconfiguration est reliée avec le configurateur par la relation de trace « trace ». De plus, les deux exigences filles « Choisir une configuration » et « Concevoir une configuration » dérivent de l'exigence mère « Rechercher une configuration » par la relation de dérivation « deriveReq » et elles sont également satisfaites par le bloc « Choix/Construction une configuration » en utilisant la relation de satisfaction « satisfy » (**Chalfoun et al., 2012a**). Pratiquement, ce bloc représente une interface qui nous permettra de choisir une configuration dans une bibliothèque de configurations déjà conçues, ou de concevoir une nouvelle configuration, ce bloc sera détaillé dans la dernière partie de ce mémoire.

D'autre part, l'exigence mère « Implanter une configuration » est dérivée de trois exigences qui sont précisées par les cas d'utilisation correspondants. Nous allons construire quelques diagrammes de séquence SysML selon le besoin dans la troisième partie « Déploiement d'un SPRA ». Tous les autres diagrammes des exigences sont détaillés et illustrés en **Annexe B**.

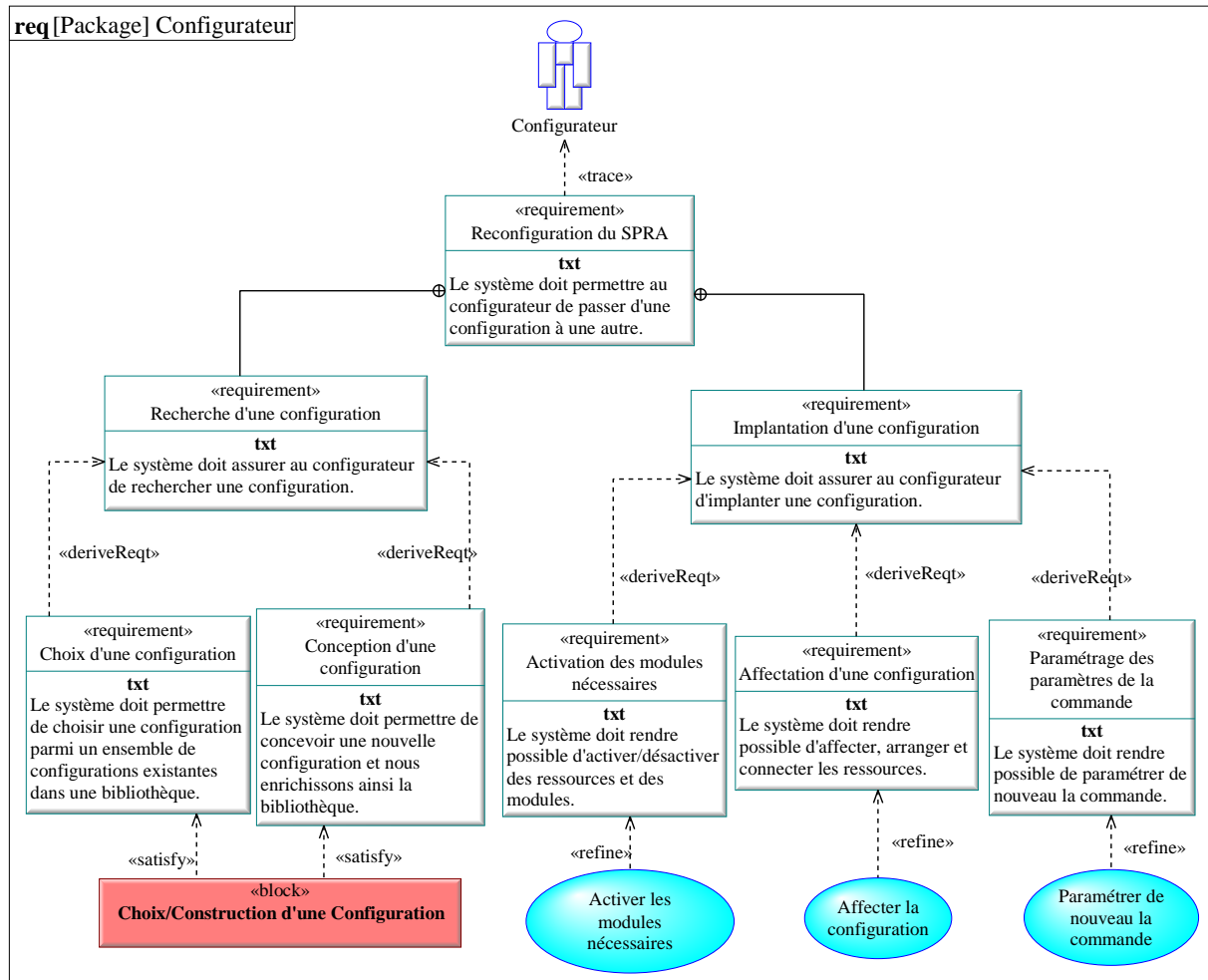


Figure 2.2.2. Diagramme détaillé des exigences du configurateur.

Nous avons utilisé les relations de traçabilité standardisées suivantes qui nous permettent de relier les exigences avec les éléments du modèle :

- « **trace** » : il s'agit d'une relation sans contrainte entre deux éléments de type quelconque.
- « **Containment** » : les exigences sont contenues dans une exigence de niveau supérieur.
- « **deriveReq** » : cette relation lie une exigence dérivée à son exigence source. Donc, elle permet d'expliciter un besoin qui était implicite dans un besoin de niveau hiérarchique supérieur de la décomposition. De plus, cette relation est utilisée uniquement entre les exigences.
- « **refine** » : c'est une relation de dépendance entre deux éléments de modèle. L'un peut être une exigence et l'autre est souvent un cas d'utilisation. En outre, cette relation sert à ajouter des détails à une exigence.

- « **verify** » : il s'agit d'une relation de dépendance qui définit comment un cas de test vérifie une exigence.
- « **satisfy** » : cette relation de dépendance décrit comment une exigence est satisfaite par un élément de modèle qui est souvent un bloc.

2.4. Allocation des exigences

Les relations de traçabilité sont clairement visibles dans les diagrammes d'exigences, mais comment pouvons-nous les voir sur les diagrammes qui contiennent des éléments pertinents du modèle ?

Le langage de modélisation SysML nous permet de réaliser l'allocation des exigences aux éléments structuraux et comportementaux du modèle. Ceci permet d'une part de voir si le modèle couvre l'ensemble des exigences et d'autre part de fournir des objectifs pour la conception détaillée des éléments du système. Les Tableaux suivants présentent des extraits d'allocations rassemblant l'ensemble des relations qui ont été déclarées entre les exigences et les éléments du modèle, en utilisant Artisan Studio (Atego, 2010) :

Name « Exigence »	Txt « Description »	Traces To « acteur Principal »	Refined By « Cas d'Utilisation »
Test à vide	Le système doit permettre de faire un test à vide.	«Actor» Chef de l'Atelier	«Use Case» Faire un test à vide
Lancement de la production	Le système doit assurer un lancement de la production.		«Use Case» Démarrer la production
Pilotage et Supervision de la production du système global	Le système doit permettre de suivre facilement et clairement le déroulement de la production du système global.		«Use Case» Suivre le déroulement de la production du système global
Arrêt de la production	Le système doit permettre d'arrêter la production.		«Use Case» Arrêter la production
Demande d'une maintenance	Le système doit rendre possible les demandes de maintenance.		«Use Case» Demander une maintenance

Tableau 2.2.2. Extrait d'allocation des exigences/Chef de l'atelier : « Traces To » et « Refined By ».

Name « Exigence »	Txt « Description »	Traces To « Acteur Principal »	Containing Class « Exigence »	Derived From « Exigence »	Refined By « Cas d'Utilisation »	Satisfied By « Bloc »
Reconfiguration du SPRA	Le système doit permettre au configurateur de passer d'une configuration à une autre.	« Actor » Configurateur				
Recherche d'une configuration	Le système doit assurer au configurateur de rechercher une configuration.		« Requirement » Reconfiguration du SPRA			
Implantation d'une configuration	Le système doit assurer au configurateur d'implanter une configuration.					
Choix d'une configuration	Le système doit permettre de choisir une configuration parmi un ensemble de configurations existantes dans une bibliothèque.			« Requirement » Recherche d'une configuration		« Block » Choix/Construction d'une Configuration
Conception d'une configuration	Le système doit permettre de concevoir une nouvelle configuration et nous enrichissons ainsi la bibliothèque.					Choix/Construction d'une Configuration
Affectation d'une configuration	Le système doit rendre possible d'affecter, arranger et connecter les ressources.			« Requirement » Implantation d'une configuration	« Use Case » Affecter la configuration	
Activation des modules nécessaires	Le système doit rendre possible d'activer/désactiver des ressources et des modules.				« Use Case » Activer les modules nécessaires	
Paramétrage des paramètres de la commande	Le système doit rendre possible de paramétrer de nouveau la commande.				« Use Case » Paramétrer de nouveau la commande	

Tableau 2.2.3. Extrait d'allocation des exigences/Configurateur: « Traces To », « Containing Class », « Derived From », « Refined By » et « Satisfied By ».

Name « Exigence »	Txt « Description »	Traces To « acteur Principal »	Refined By « Cas d'Utilisation »	Satisfied By « Bloc »
Début du travail sur un poste	Le système doit permettre de commencer le travail individuellement sur chaque poste.	«Actor» Opérateurs		«Block» Commence du travail sur un poste
Production	Le système doit permettre de produire selon la gamme logique.		«Use Case» Produire	
Suivi du déroulement de la production sur un poste	Le système doit permettre de suivre le déroulement de la production individuellement sur chaque poste.		«Use Case» Suivre le déroulement de la production sur un poste	
Gestion de la performance	Le système doit rendre possible à l'opérateur de gérer la performance de la production, de la qualité et de la maintenance.			«Block» Indicateurs de la Performance
Clôture du travail sur un poste	Le système doit rendre possible aux opérateurs de finir le travail.			«Block» Cloture du travail sur un poste
Demande d'une maintenance	Le système doit rendre possible aux opérateurs de demander une maintenance.			«Block» Demande d'une maintenance

Tableau 2.2.4. Extrait d'allocation des exigences/Opérateurs: « Traces To », « Refined By » et « Satisfied By ».

Name « Exigence »	Txt « Description »	Traces To « acteur Principal »	Derived From « Exigence »	Satisfied By « Bloc »
Maintenance du système	Le système doit permettre de maintenir facilement ses différents constituants.	«Actor» Opérateurs de la Maintenance		
Maintenance en cas de demande	Le système doit assurer une maintenance en cas de demande.		«Requirement» Maintenance du système	«Block» Maintenance
Maintenance régulière	Le système doit assurer une maintenance régulière.			

Tableau 2.2.5. Extrait d'allocation des exigences/Opérateurs de la Maintenance: « Traces To », « Derived From » et « Satisfied By ».

En effet, le block est la brique de base dans la déclaration des concepts constitutifs d'un système. Dans le chapitre précédent, nous avons détaillé quelques blocs en utilisant le diagramme de définition de blocs (BDD : Block Definition Diagram), la décomposition des éléments complexes est parfaitement possible avec le diagramme interne de bloc (IBD : Internal Block Diagram) qui décrit la structure interne de l'élément en termes de parties, ports et connecteurs.

Les exigences principales de la reconfigurabilité et de l'agilité du système étudié impliquent que les différentes ressources (postes de travail, robots,...) doivent être modulaires, facilement déplaçables, et également facilement connectées et câblées. Donc, il faut bien traduire ces exigences en capacités des ressources et des communications entre les différents constituants du SPRA en tenant compte des différents types d'interfaces (interface de l'énergie, interface pour recevoir les pièces, interface de communications,...).

Dans le but de bien caractériser la fonction de reconfiguration d'un SPRA, nous aurons besoin de représenter les flots de données et de contrôle entre les actions. Le diagramme d'activité est l'un des diagrammes dynamiques proposés par SysML, il peut faire le lien avec les blocs de la modélisation structurelle et peut aussi modéliser des flux continus. Il est utilisé majoritairement pour l'expression de la logique de contrôle et d'entrées/sorties. Nous allons ainsi construire quelques diagrammes d'activités selon le besoin dans la troisième partie « Déploiement d'un SPRA ».

Pratiquement, pendant la validation de ces concepts, de nouvelles exigences peuvent être générées et prises en compte au niveau « système » ou « sous système » parce que la spécification des exigences est un processus continu et itératif. Les exigences servent ainsi à établir un contrat entre le client qui forme le cahier de charges et les réalisateurs du futur système qui rédigent les exigences techniques.

3. Description de haut niveau des fonctionnalités du SPRA

Un modèle de cas d'utilisation fournit une description de haut niveau des fonctionnalités du système et il montre également les interactions entre les acteurs extérieurs au système et les fonctionnalités du système. Ainsi, il représente les services attendus du système à l'étude.

Dans ce contexte, le langage de modélisation des systèmes complexes SysML nous propose de définir le diagramme de cas d'utilisation qui peut être défini à plusieurs niveaux pour faciliter et servir l'étude du système et sa documentation.

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel, ...) qui interagit directement avec le système étudié. Un acteur participe au moins à un cas d'utilisation. En plus, nous distinguons les acteurs principaux des acteurs secondaires. Les acteurs principaux ou humains cherchent à obtenir un résultat observable ou un but en utilisant les services du système. Par

contre, les acteurs secondaires sont généralement les acteurs non humains comme les équipements. En effet, cela sert à clarifier les besoins par rapport aux services rendus à différents profils d'utilisateurs et également à identifier les éventuels systèmes externes sollicités pour rendre les services.

Pour notre système SPRA, nous identifions ainsi quatre types d'acteurs principaux qui sont « le configurateur », « le chef de l'atelier », « les opérateurs » et « les opérateurs de la maintenance », et également quatre types d'acteurs secondaires qui sont « les ressources stationnaires », « les ressources de transport », « le produit » et « l'ordinateur de supervision ».

En outre, un cas d'utilisation (UC : Use Case) représente un ensemble de scénarios d'utilisation d'un système reliés par un but commun du point de vue d'un acteur principal. Un scénario est une séquence particulière d'actions s'exécutant du début à la fin du cas d'utilisation. En général, un cas d'utilisation contient un scénario nominal et/ou plusieurs scénarios alternatifs qui se terminent de façon normale, et également des scénarios d'erreur qui se terminent en échec. En d'autres termes, un cas d'utilisation représente un ensemble de séquences d'actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier. Chaque cas d'utilisation permet de décrire ce que le système devra faire, sans spécifier comment il le fera. De plus, chaque cas d'utilisation doit être relié à au moins à un acteur. D'autre part, un cas d'utilisation ne doit pas se réduire systématiquement à une seule séquence. Il est conseillé donc de rester synthétique et de ne pas tomber dans le piège de la granularité trop fine des cas d'utilisation. Par conséquent, les cas d'utilisation servent à décrire le fonctionnement du système de façon à être partagé avec un client ou une maîtrise d'ouvrage, à exprimer des exigences fonctionnelles qui seront validées avec des utilisateurs finaux, à clarifier les échanges avec l'environnement, et également à identifier les premières dépendances fonctionnelles.

Dans ce cas de SPRA, nous définissons les cas d'utilisation principaux du système qui sont « Rechercher une configuration », « Implanter une configuration », « Démarrer, Suivre et Arrêter la production », « Produire » et « Maintenir les parties du système » (**Holt *et al.*, 2008**), (**Roques, 2009**).

Dans cette optique, une description de haut niveau des fonctionnalités du système (Figure 2.2.3), en utilisant Artisan Studio, permet de montrer les relations entre les acteurs principaux et secondaires d'une part, et les cas d'utilisation principaux d'autre part. Généralement, les acteurs principaux sont à gauche des cas d'utilisation et les acteurs secondaires sont à droite.

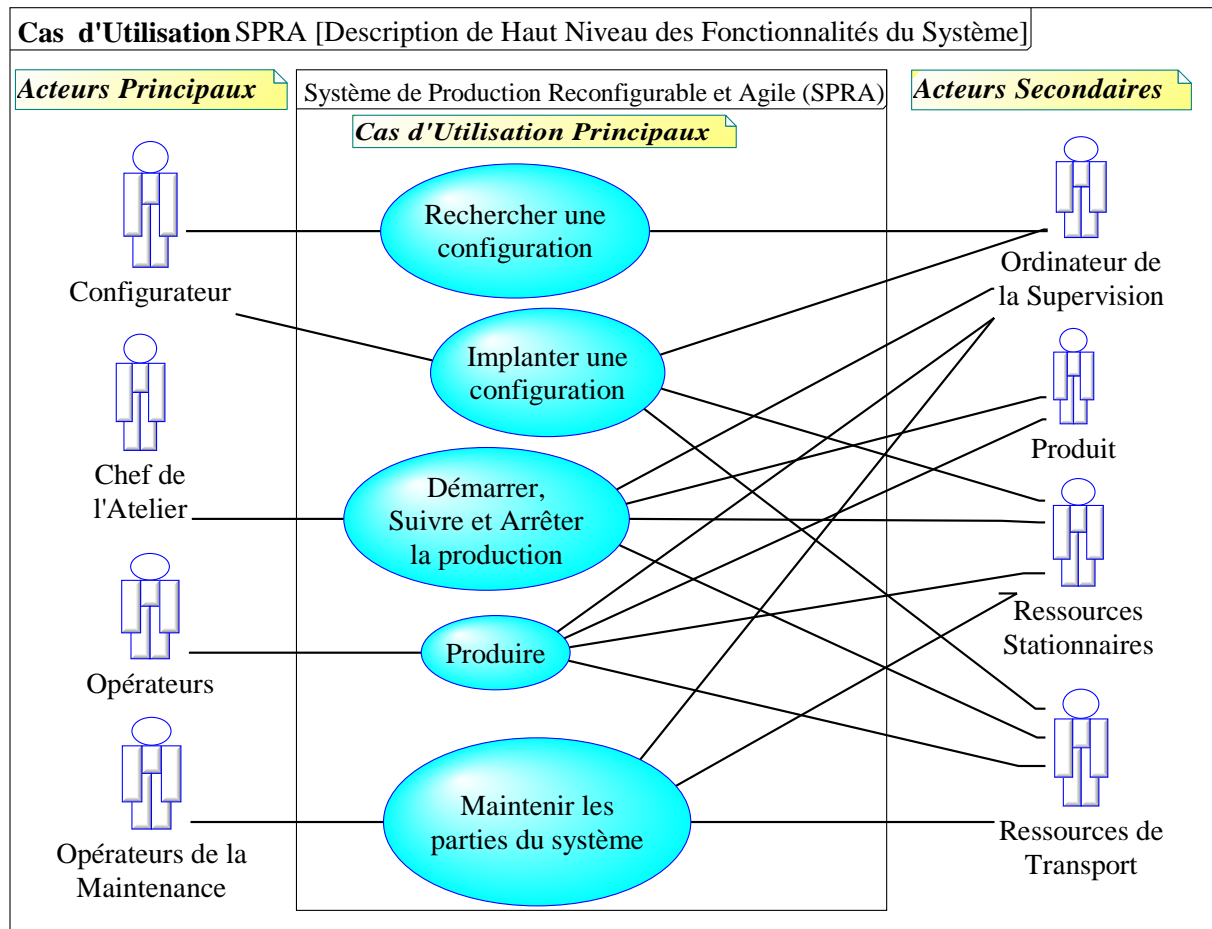


Figure 2.2.3. Diagramme de cas d'utilisation d'un SPRA.

Cependant, pour tout projet d'ingénierie complexe comme le SPRA, ce diagramme est susceptible de contenir un grand nombre de cas d'utilisation, ce qui rend très difficile la compréhension du contexte du projet. Une meilleure approche consiste à construire des diagrammes de cas d'utilisation opérationnels pour les acteurs principaux. Pour cela, nous identifions les objets et les buts de chaque acteur principal. Par exemple, la Figure 2.2.4 illustre les cas d'utilisation opérationnels du configurateur qui est le concepteur du système.

Quand nous avons besoin de reconfigurer le système, le configurateur recherche une nouvelle configuration et l'implante. Cette nouvelle configuration doit répondre aux besoins du système. Donc, le configurateur peut choisir une configuration parmi un ensemble de configurations existantes dans une bibliothèque. Si aucune configuration ne convient, il conçoit une configuration et nous enrichissons ainsi la bibliothèque. Afin que le configurateur puisse planter la nouvelle configuration, il est nécessaire d'affecter la configuration, paramétrer de nouveau la commande et activer les modules nécessaires.

Nous avons illustré cela sur la Figure 2.2.4 en reliant les acteurs et les cas d'utilisation par les trois types de relations standardisées qui sont (**Holt et al., 2008**) :

➤ La relation d'inclusion formalisée par le mot-clé « **include** » : le cas d'utilisation de base contient explicitement l'autre de façon obligatoire. Cette relation est utilisée quand une sous-fonction peut être une partie du cas d'utilisation principal. Autrement dit, le cas d'utilisation inclus est considéré comme une partie du cas d'utilisation de parent.

La direction de la flèche doit donner un sens lorsque le modèle est lu. Par exemple, cette partie du modèle est lue comme « Rechercher une configuration » inclut « Choisir une configuration » et « Concevoir une configuration ». Ainsi, ces deux cas d'utilisation sont déclenchés systématiquement par le cas d'utilisation qui l'inclut.

➤ La relation d'extension formalisée par le mot-clé « **extend** » : le cas d'utilisation étendu est un comportement optionnel du système à partir du cas d'utilisation de base. En d'autres termes, le cas d'utilisation étendu fait parfois partie du cas d'utilisation de parent.

Par contre, il est important de tracer correctement la direction de cette relation car elle est différente de la direction « include ». Par exemple, le cas d'utilisation « Choisir une configuration » déclenche sous conditions le cas d'utilisation « Concevoir une configuration ». Ainsi, ces deux relations « include » et « extend » servent à factoriser des scénarios.

➤ La relation de **généralisation** (flèche blanche) : le cas d'utilisation fils hérite le comportement et les propriétés de son père. Chacun d'entre eux peut néanmoins comprendre des interactions spécifiques supplémentaires. Donc, il s'agit d'un mécanisme par lequel on « factorise » des caractéristiques communes à plusieurs entités dans une entité générique. Cette relation sert à « alléger » les diagrammes et à définir des « patterns » génériques réutilisables dans différents contextes.

La Figure 2.2.4 montre que le cas d'utilisation principal « Implanter une configuration » a deux types : « Choisir une configuration » et « Concevoir une configuration ». De plus, le cas d'utilisation « Implanter une configuration » inclut trois cas d'utilisation de niveau inférieur : « Affecter la configuration », « Paramétrer de nouveau la commande » et « Activer les modules nécessaires ». En outre, le cas d'utilisation « Activer les modules nécessaires » peut être étendu par le cas d'utilisation « Paramétrer de nouveau la commande » (**Chalfoun et al., 2012a**).

En raison du fait que « Choisir une configuration » et « Concevoir une configuration » sont deux types de « Implanter une configuration », ils comprennent aussi les trois cas d'utilisation « Affecter la configuration », « Paramétrer de nouveau la commande » et « Activer les modules nécessaires ».

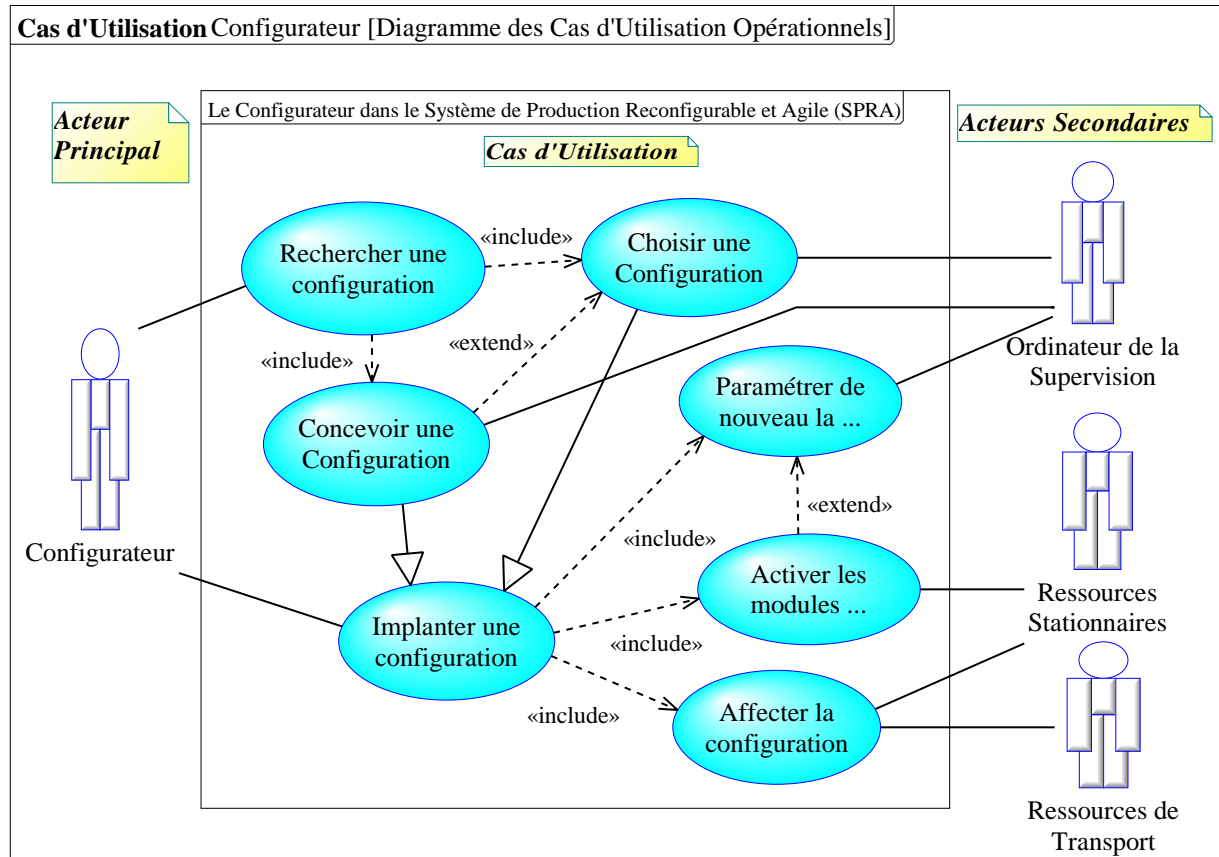


Figure 2.2.4. Diagramme de cas d'utilisation opérationnels du configureur.

Les trois autres diagrammes de cas d'utilisation opérationnels sont détaillés et illustrés en **Annexe C**.

En conséquence, cette description textuelle et graphique est indispensable pour documenter les cas d'utilisation de la reconfiguration d'un SPRA car elle permet de communiquer facilement avec les utilisateurs. En revanche, le texte présente des désavantages puisqu'il est difficile de montrer comment les enchaînements se succèdent, ou à quel moment les acteurs secondaires sont sollicités. Il sera donc recommandé, selon le besoin, de compléter cette description textuelle par un ou plusieurs diagrammes de séquence SysML. Le diagramme de séquence montre la séquence verticale des messages passés entre éléments (lignes de vie) au sein d'une interaction. Nous considérons le système entier comme une boîte noire et le représentons par une seule ligne de vie. Le comportement du système est décrit de l'extérieur, sans montrer les interactions internes. Nous allons construire quelques diagrammes de séquence selon le besoin dans la troisième partie « Déploiement d'un SPRA ».

Dans la section suivante, nous allons introduire une description structurale du SPRA qui représente actuellement une application du laboratoire.

4. Description structurelle du SPRA

Le SPRA peut être représenté comme un système industriel constitué de ressources stationnaires, k postes de travail déplaçables PT_1, \dots, PT_k (par exemple : usinage, traitement, assemblage, ...), q postes de secours déplaçables PS_1, \dots, PS_q qui permettent au système de continuer à fournir des services lorsqu'un poste tombe en panne, deux tampons IN et OUT qui sont respectivement les buffers d'entrée et de sortie, et d'un ordinateur de supervision. Chaque poste est équipé d'un casier de stockage placé à son entrée. Toutes ces ressources stationnaires peuvent être accessibles via un robot AGV (Automated Guided Vehicle). Les tampons IN et OUT sont associés aux ressources de transport « chargeur » et « déchargeur », les fonctions de transport sont TF_{in} et TF_{out} respectivement. Chaque poste de travail PT_k est associé à une ressource de transport Tr_k et à une fonction de transport TF_k qui doit être définie.

Supposons que nous ayons n fonctions de travail définies sur ce système F_1, \dots, F_n . Ces fonctions interviennent dans la réalisation de j produits Pr_1, \dots, Pr_j décrits par les gammes logiques qui sont définies comme des séquences de fonctions nécessaires, par exemple : $G_1 (F_1, F_3, \dots), \dots, G_j (F_2, F_5, \dots)$.

Les produits bruts entrent par IN où ils peuvent être stockés, les produits finis sont évacués sur OUT qui fournit aussi une capacité de stockage.

Les opérations sont décomposées en six groupes détaillés dans le chapitre précédent : les opérations principales, les opérations de transports, les opérations de stockage, les opérations d'activation, les opérations d'affectation et les opérations d'implantation. Pour les opérations principales, elles associent une ressource stationnaire à une fonction de travail $Op_1 (PT_1, F_1), \dots, Op_r (PT_k, F_n)$ dont les temps opératoires sont respectivement t_1, \dots, t_r . Chaque fonction peut alors être implémentée par une ou plusieurs opérations. Par exemple, F_1 est implémentée par 2 opérations réalisées sur PT_1 et PT_3 , nous définissons alors les deux opérations $Op_1 (PT_1, F_1)$ et $Op_2 (PT_3, F_1)$. En plus, les opérations de transports sont reliées aux connexions et aux instances de fonctions de transport correspondantes. En ce qui concerne les opérations de stockage, elles relient chaque tampon à un produit $St_0 (Pr_0, IN), \dots, St_j (Pr_j, OUT)$.

Chaque produit Pr_j est normalement décrit par une gamme logique G_j qui génère une configuration. Pratiquement, le nombre minimum de machines (N) nécessaires dans le système est calculé par l'équation (1) dans le chapitre 2 de la première partie. Nous déterminons ensuite le nombre des phases (m) et le nombre de postes à chaque phase $np[m]$ selon les fonctions de travail décrites par la gamme logique du produit. Nous avons déjà présenté une méthode mathématique pratique qui nous permet de calculer le nombre de configurations possibles de type RMS. Par définition, une configuration du système est une organisation temporaire qui prend fin soit parce que les objectifs fixés sont atteints, soit parce que les contraintes appliquées au

système ne sont plus respectées. Une décision multicritère sera prise par le configurateur pour choisir une configuration idéale. Nous pouvons citer quelques critères qui peuvent être considérés pour évaluer les configurations et pour prendre ensuite la décision : le nombre total de postes, le nombre de phases, le nombre de postes à chaque phase, l'arrangement de postes et les connexions entre eux, le coût de production qui concerne notamment la consommation d'énergies et d'équipement et d'outillage, et bien sûr le coût de la reconfiguration. Une bonne configuration est celle qui respecte les délais de travail entre les postes dans chaque phase et entre les phases elles-mêmes.

Nous donnons ainsi les coordonnées cartésiennes (x, y) pour chaque poste au niveau de l'atelier représenté par un repère absolu. Les connections et les ressources de transport associées aux postes dans cet arrangement devront être aussi définies. Le chef de l'atelier et les opérateurs réaliseront cette nouvelle configuration soit à l'aide du robot AGV, soit manuellement.

La question la plus importante concerne l'identification des informations optimisées, indispensables et minimums, pour faire interagir les composants modulaires du SPRA. Quelles sont les interactions des postes entre eux, entre chaque poste et l'ordinateur de supervision, entre chaque poste et le produit, entre l'ordinateur de supervision et les ressources de transport et les opérateurs et le robot AGV, etc. Effectivement, nous pensons que les réponses à toutes ces questions serviront et fourniront l'agilité à notre système.

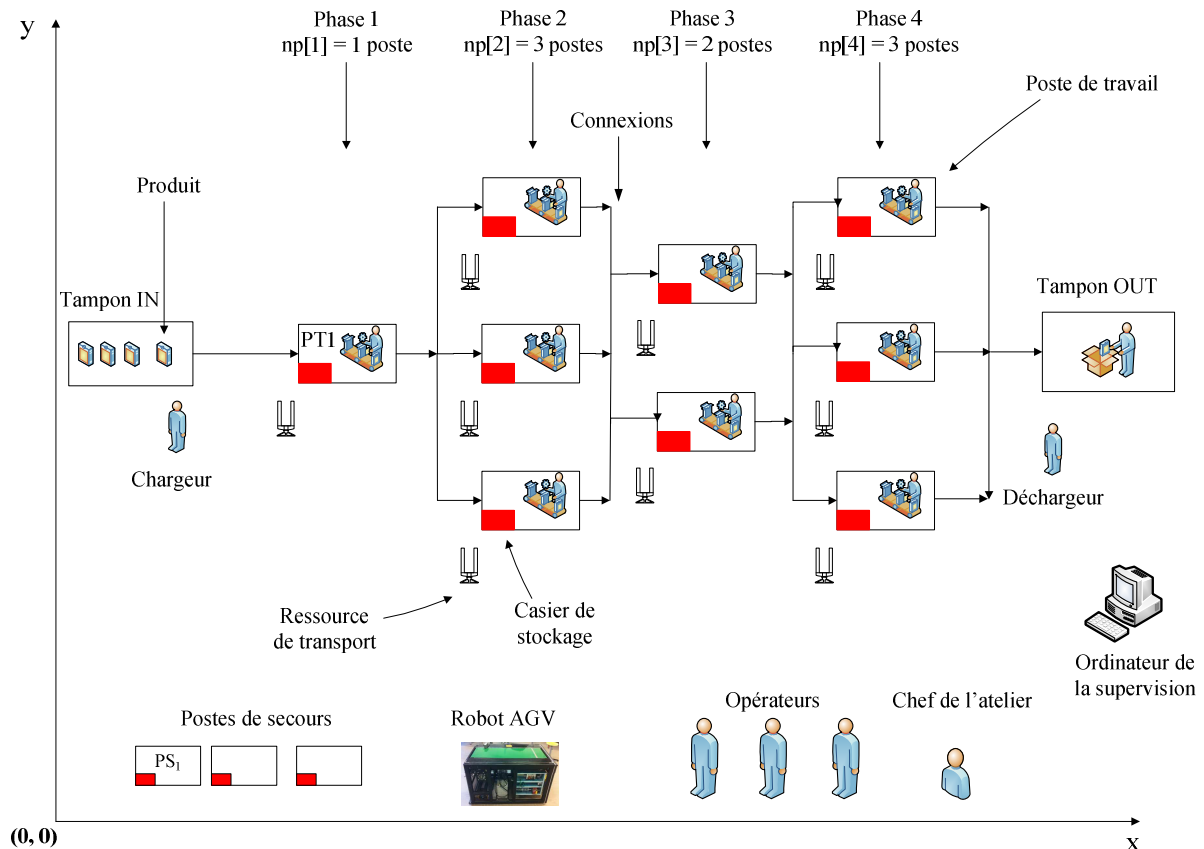


Figure 2.2.5. Vue globale de l'atelier.

La Figure 2.2.5 présente un schéma détaillé de la structure proposée de ce SPRA et un exemple de configuration de type RMS ($N = 9$ et $m = 4$) avec des connexions croisées après chaque phase. Donc, un produit de n'importe quel poste à la phase (i) peut être transféré à n'importe quel poste à la phase ($i + 1$). Tous les postes et les opérations sont identiques à chaque phase.

Effectivement, toutes les exigences déjà proposées peuvent être représentées et réalisées potentiellement à l'aide des différents constituants du système et des connexions d'une part, et avec l'environnement de l'atelier d'autre part (Figure 2.2.6).

SPRA: Système de Production Reconfigurable et Agile

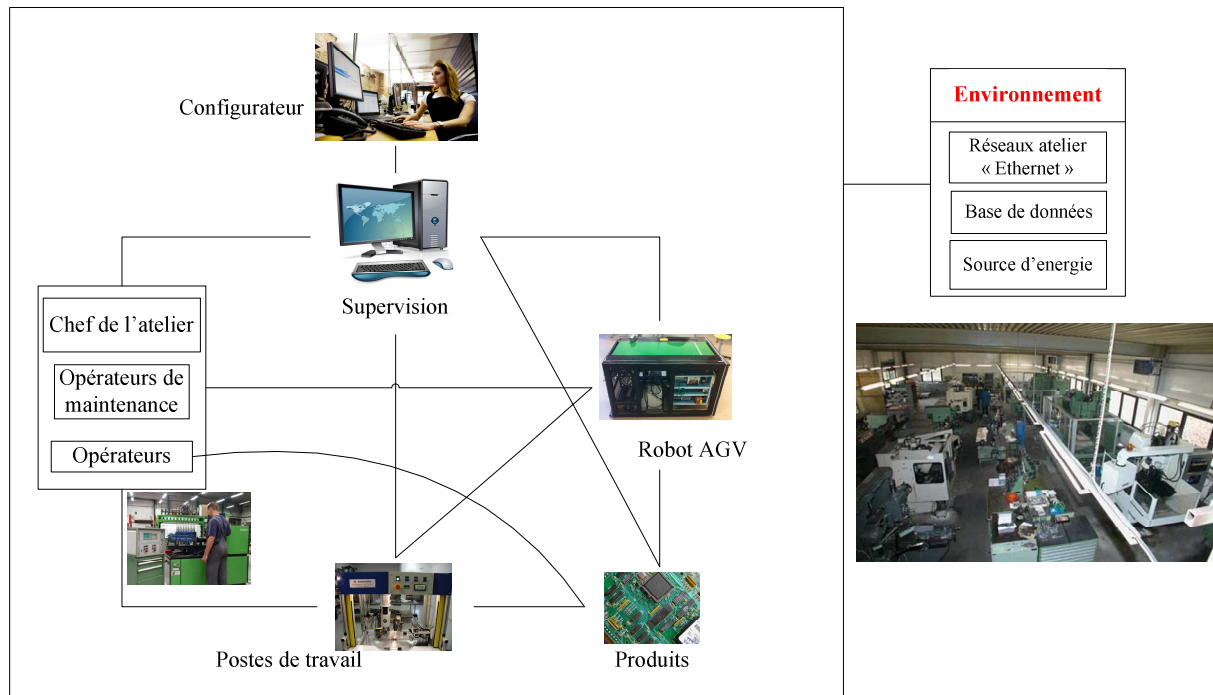


Figure 2.2.6. Contexte organique d'un SPRA.

5. Conclusion

Dans ce chapitre, nous avons spécifié la fonction de reconfiguration. Ainsi, nous avons analysé et défini les exigences principales du SPRA. De plus, une allocation des exigences aux éléments structuraux et comportementaux du modèle a été réalisée en définissant les blocs permettant de satisfaire un certain nombre d'exigences, et les cas d'utilisation permettant de raffiner quelques exigences.

Ensuite, nous avons présenté une description de haut niveau des fonctionnalités et de la structure du futur système SPRA en se focalisant sur la fonction de reconfiguration. Les acteurs et les cas d'utilisation principaux ont été identifiés.

Après avoir proposé dans cette partie un modèle générique adapté à ce futur système et également une caractérisation du SPRA, nous allons introduire l'aspect de déploiement d'une nouvelle configuration. Le processus de reconfiguration, les diagrammes nécessaires des séquences et des activités, et l'architecture de commande au niveau système seront l'objectif de la partie suivante.

Troisième Partie

Déploiement d'un SPRA

Cette partie a pour objectif d'apporter des réponses à la question « comment déploie-t-on un SPRA dans une certaine configuration ? ».

Tout d'abord, une méthode pour réaliser la fonction de reconfiguration sera proposée. Ainsi, un processus de reconfiguration représentant une démarche à suivre pour concevoir et implanter une nouvelle configuration sera développé afin de répondre aux nouveaux besoins du client et/ou d'assurer une continuité du service avec une nouvelle organisation du système. De plus, une modélisation des opérations de reconfiguration sera proposée sous forme de diagrammes des séquences et de diagrammes des activités SysML.

De même, un pilotage opérationnel compatible avec le SPRA sera présenté dans cette partie. D'abord, nous proposerons une architecture de pilotage adaptée au SPRA (APHA-SPRA). Ensuite, nous définirons les interactions entre les différents composants du système de pilotage représentant le fonctionnement opérationnel du SPRA.

Sommaire

1. Introduction	153
2. Processus de reconfiguration.....	154
3. Modélisation des opérations de reconfiguration	160
3.1. Diagrammes d'activité	160
3.2. Diagrammes de séquence « système »	162
4. Architecture de Pilotage Hybride Adaptée au SPRA (APHA-SPRA).....	166
5. Fonctionnement opérationnel du SPRA	169
6. Conclusion	175

1. Introduction

Un système de production peut être défini comme un ensemble d'opérateurs, de matériels (machines, outils, matières) et de logiciels (programmes de commande) qui sont intégrés pour effectuer une série contrôlée d'opérations répétitives sur les matières premières. L'objectif du système est alors d'obtenir une forme finale souhaitée ou d'assembler un produit final répondant aux besoins de clients. Systématiquement, chaque système de production est caractérisé par son aspect statique (sa structure : ses composants matériels), son aspect dynamique (processus de production) et son mode d'exploitation (NASA, 1995), (INCOSE, 2004), (AFIS, 2009). Par ailleurs, une configuration du système est une organisation temporaire de l'ensemble des caractéristiques matérielles et logicielles du système. Chaque configuration est définie d'une façon à répondre à la gamme logique de produit planifiée et elle peut prendre fin si nécessaire.

Effectivement, le système d'information gère efficacement les données : collecte et traitement automatisés et mise à disposition dans un temps compatible avec les besoins « temps réel ». De plus, il réduit les risques : amélioration de la flexibilité, de la compétitivité, adaptation plus simple aux changements. Il optimise également le coût total de possession. Donc, une bonne information est un préalable pour de bonnes décisions. La qualité de l'information réside autant dans son contenu qui doit être adapté à l'usage que dans la façon dont elle est obtenue et diffusée. Sans aucun doute, le personnel doit aussi être attentif et intéressé aux performances de la production.

Dans cette partie, nous introduisons une nouvelle méthodologie qui explique comment déploie-t-on une configuration. D'abord, nous présentons les mécanismes mis en œuvre lors du changement de configuration qui sont appelées « processus de reconfiguration ». Ce processus permet de réaliser un nouvel objectif ou de pallier une défaillance lors de fonctionnement du système en choisissant/construisant une nouvelle configuration et en la mettant également en place. Effectivement, le but du processus de reconfiguration est de répondre aux besoins du client et d'assurer une continuité du service avec une nouvelle organisation du système. Ensuite, nous modélisons les différentes étapes de reconfiguration sous forme de diagrammes de séquence et de diagrammes d'activité SysML.

Egalement, nous proposons une architecture de pilotage compatible avec ce nouveau type de système de production (APHA-SPRA). Une architecture globale de commande sera ensuite présentée. Enfin, nous définissons les composants principaux du système de pilotage et nous expliquons les différentes interactions entre eux dans le but de réaliser un pilotage opérationnel pratique et efficace pour un système de production reconfigurable et agile (SPRA).

2. Processus de reconfiguration

La reconfiguration d'un système de production peut être définie comme la modification de ses composants matériels (ajouter/annuler des postes, réactiver leurs capacités) et/ou de ses composants logiciels (réimplanter les programmes de contrôle) et/ou de son mode d'exploitation afin d'ajuster la capacité et la fonctionnalité de la production en répondant aux changements brusques du marché et/ou aux exigences régulières.

Dès que la configuration actuelle du système prend fin soit parce que les objectifs fixés sont atteints (la quantité du produit actuel est complète), soit parce que les contraintes appliquées au système ne sont plus respectées (par exemple : une défaillance, une panne, un changement de production en cas d'un arrêt partiel ou total de la production et en cas d'un nouveau produit), un processus de reconfiguration doit démarrer pour mettre en place une nouvelle configuration capable de réagir efficacement aux nouvelles exigences.

Différents niveaux de reconfiguration peuvent alors être considérés à leur tour. Ces niveaux vont de la réorganisation des opérations déjà utilisées à la modification complète de la structure du système qui correspond à l'ajout, le remplacement, la suppression ou la modification d'une ressource. En d'autres termes, la nouvelle configuration doit consister à utiliser différemment les opérations mises en œuvre actuellement, nous parlons dans ce cas d'une reconfiguration mineure (la même configuration actuelle), soit toutes les potentialités du système sont changeables, nous avons donc une reconfiguration majeure (une nouvelle configuration). Dans les deux cas, lors du changement de configuration, nous avons besoin de définir les mécanismes qui réalisent la fonction de la reconfiguration expliquant les étapes à suivre pour passer d'une configuration à une autre. Probablement, ce passage peut nécessiter une phase transitoire parce que l'état des produits présents sur le système n'est pas nécessairement conforme à la nouvelle configuration. Dans ces travaux de thèse, nous ne traitons pas cette phase transitoire.

La fonction de la reconfiguration consiste principalement en la planification de la gestion de production et au diagnostic du système à construire.

Au niveau de la planification de la gestion de production, une reconfiguration sera demandée par le configurateur lorsque la quantité du produit actuel est complète ou en cas d'un changement de production (un arrêt partiel ou total de la production, un nouveau produit,...).

En ce qui concerne le diagnostic du système, une analyse permanente est nécessaire des indicateurs de la performance du système qui sont calculés à partir des paramètres d'entrée/sortie du système à conduire. En cas d'anomalies de moyens de production (défaillance ou panne), une information arrive au configurateur pour déclencher le processus de reconfiguration.

Dans tous les cas, le configurateur prend à son tour la décision pour reconfigurer le système selon le cas.

Le principe général de reconfiguration proposé dans ces travaux de thèse touche plusieurs étapes à suivre (Figure 3.1) en prenant compte plusieurs aspects :

- La possibilité d'introduire des nouveaux produits ;
- La planification de production : variété de produits, taux de production, maintenance, etc ;
- Les capacités des postes de travail et leurs états, l'arrêt partiel ou total de la production (statut des opérateurs, défaillance d'un ou plusieurs postes de travail ou en cas de panne,...).

L'ensemble de ces paramètres joue un rôle important dans la prise de décision pour reconfigurer le système de production au même titre que le temps nécessaire pour aboutir à la nouvelle configuration afin de répondre à la question « Quand doit-on reconfigurer un système de production ? ».

Lorsque le processus de reconfiguration est déclenché, il faut chercher une nouvelle configuration. Cette nouvelle configuration doit répondre aux différents besoins. Il est donc important de bien évaluer l'ensemble des configurations déjà conçues et enregistrées dans une bibliothèque afin de choisir la meilleure configuration qui pourrait atteindre l'objectif visé. Si aucune configuration ne convient, le configurateur conçoit une nouvelle configuration qui viendra enrichir la bibliothèque. En plus, une évaluation de cette nouvelle configuration peut être effectuée pour un retour d'expérience dans le futur. Il s'agira donc de répondre à la question « Pour aboutir à quelle configuration ? ».

Enfin, pour que le configurateur puisse implanter la nouvelle configuration, il est nécessaire de préciser les ressources capables d'intervenir dans cette configuration, de déplacer les postes de travail, et encore d'activer des modules (outils et/ou dispositifs) sur les ressources utilisées selon les exigences de la configuration afin de réimplanter la commande. Cela répond à la question « Comment reconfigure-t-on un système de production ? ».

Dans ces travaux de thèse, nous traitons uniquement la conception et l'implantation d'une nouvelle configuration. Il est évident que la partie optimisation qui permet de choisir la configuration optimale est nécessaire pour la reconfiguration mais ne sera pas traitée dans la suite des travaux. Par contre, nous proposons une modélisation du système de production et du processus de reconfiguration afin de pouvoir ajouter facilement un module d'optimisation, ou d'aide à la décision pour le choix optimal de la configuration cible.

Par conséquent, la fonction de reconfiguration est composée de deux étapes principales :

- ✓ La première consiste à choisir/concevoir une nouvelle configuration ;

✓ La seconde est consacrée à l'implantation de cette configuration en faisant éventuellement intervenir le pilotage et la gestion des modes afin de mettre le système et les produits dans un état compatible avec elle.

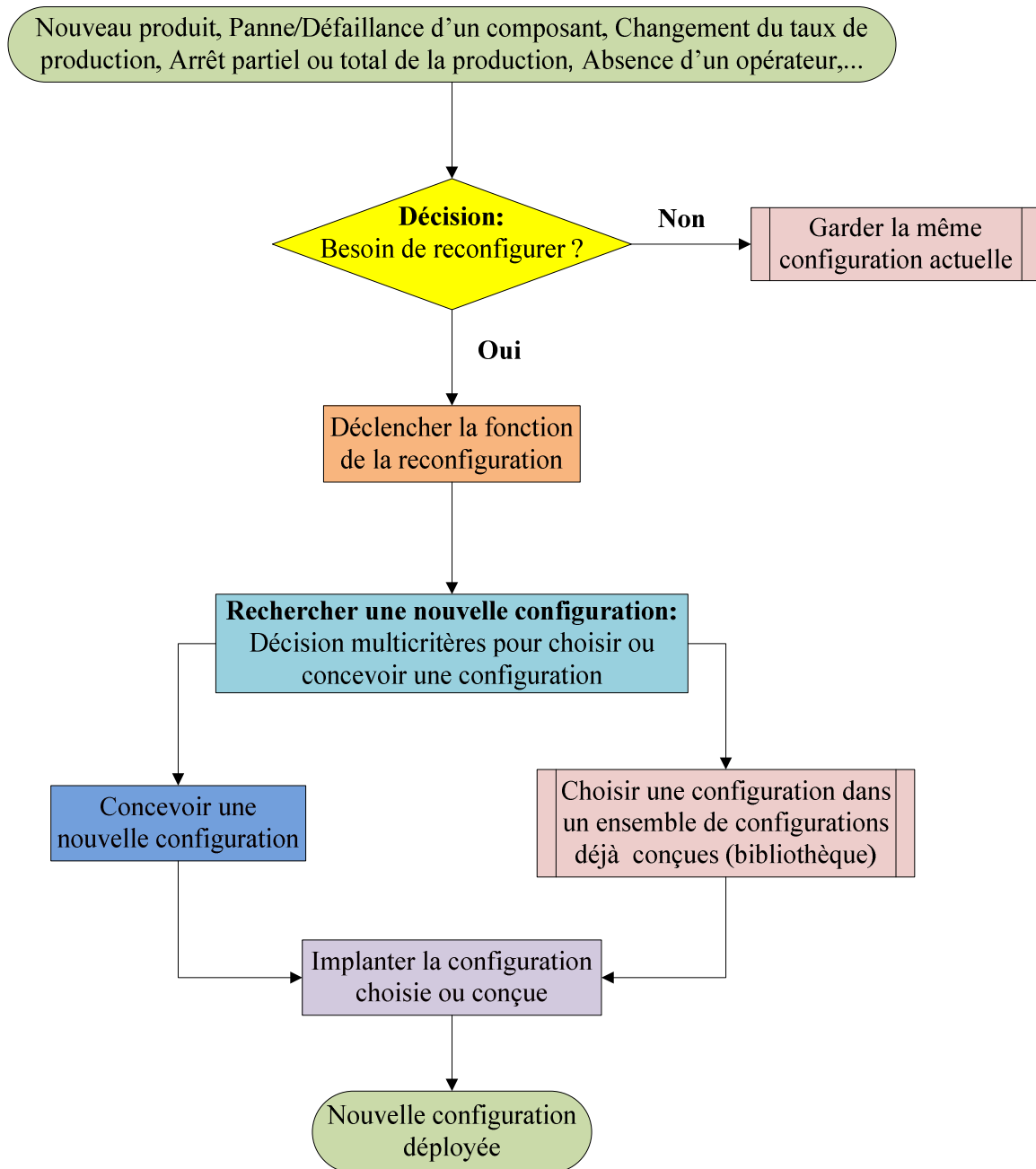


Figure 3.1. Principe général de reconfiguration du système.

De même, nous pouvons représenter ce principe général de la reconfiguration d'une façon fonctionnelle dynamique équivalente (Figure 3.2).

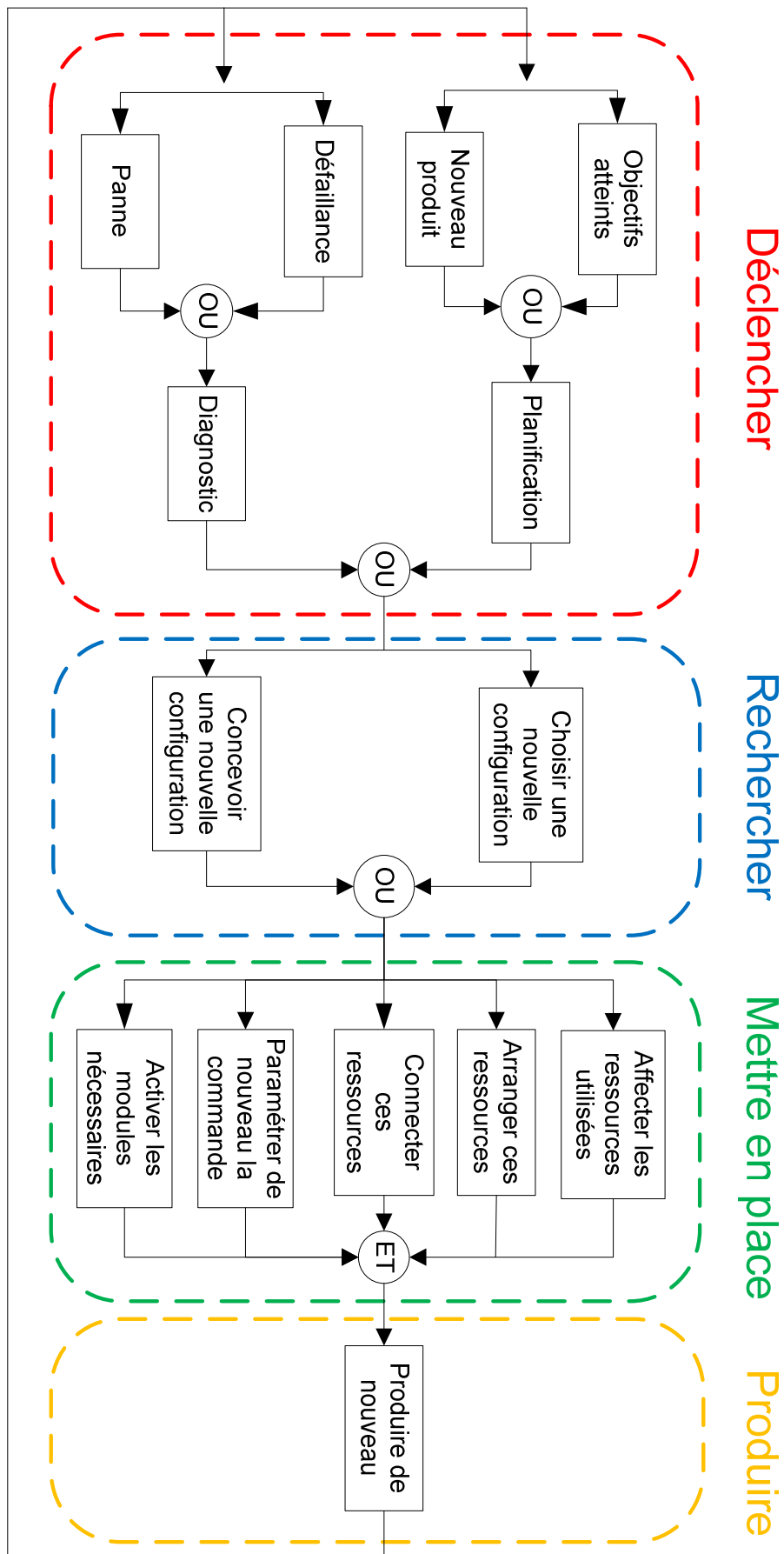


Figure 3.2. Représentation fonctionnelle dynamique de la reconfiguration du système.

Tandis que notre objectif est de concevoir un SPRA d'une manière modulaire, nous définissons alors les composants modulaires principaux : planification de la gestion de production, diagnostic de système, décision multicritères sur la reconfiguration, génération d'une nouvelle configuration, réimplantation de la commande, activation des modules et gestion des modes des ressources. En effet, ces composants servent le système de pilotage et seront donc détaillés dans la suite de ce chapitre.

Dans le cadre de l'intégration du processus de reconfiguration avec le modèle générique déjà proposé dans le premier chapitre de la deuxième partie, la fonction de la reconfiguration démarre toujours par le besoin de lancer un nouveau produit (Figure 3.3). Donc, il faut :

- Définir les fonctions qui doivent être réalisées selon la gamme logique du produit : collage, pliage, perçage, contrôle, pointage, etc.
- Définir les composants matériels capables de réaliser ces fonctions.
- Mettre en place une plateforme de réalisation, nous choisissons alors un ensemble de composants et nous les agençons. La configuration à trouver est donc constituée de :
 - ✓ Un ensemble de composants : postes de travail, convoyeurs, robots, logiciels de contrôle, etc.
 - ✓ Un agencement de ces composants entre eux : par exemple une structure linéaire, parallèle ou une combinaison de ces deux.
- Enfin, nous parlons de deux aspects de configuration :
 - ✚ Une configuration logique (logicielle) : nous affectons un programme de contrôle à chaque ressource utilisée dans cette configuration.
 - ✚ Une configuration physique (matérielle) de chaque poste : nous branchons par exemple une visseuse dans un poste prévu à cet effet.

En outre, nous rappelons que les opérations principales de la gamme logique associent les fonctions de travail aux postes de travail et il faut bien noter que plusieurs opérations principales peuvent être réalisées sur un même poste. Par exemple, une opération de perçage et une opération de contrôle sont affectées au même poste qui doit en être capable.

De plus, la Figure 3.4 montre comment les différentes parties de ce modèle générique sont reconfigurées. Effectivement, nous commençons toujours par la structure logique (les produits et leurs gammes logiques) qui reconfigure le système et définit la structure physique (les ressources et les différentes connexions entre eux). Parallèlement et selon la structure physique, cette structure logique reconfigure le système et définit la configuration logique (les programmes de contrôle) qui, à son tour, reconfigure le système et définit la configuration physique (les ressources utilisées et les capacités demandées de chaque ressource).

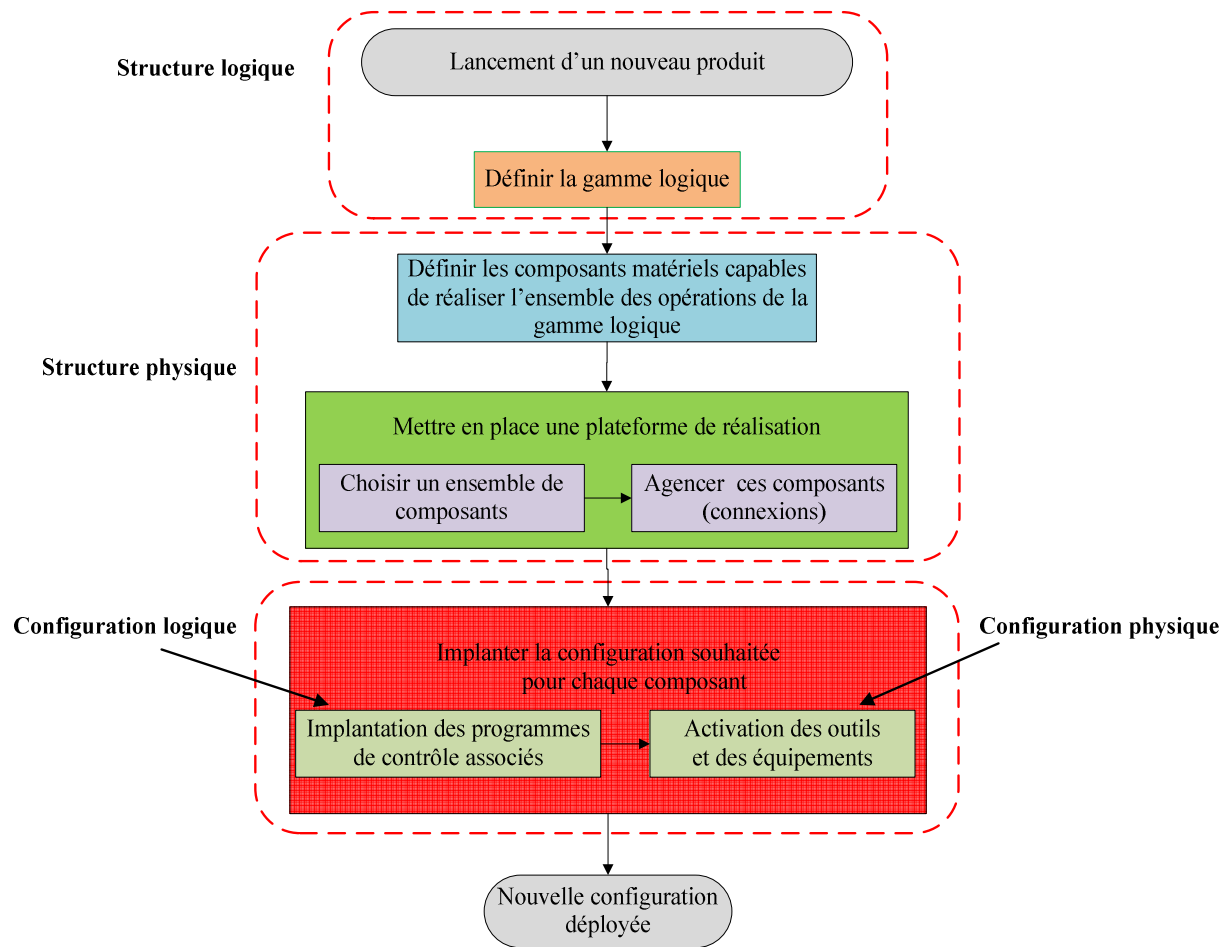


Figure 3.3. Processus de conception et implantation d'une nouvelle configuration.

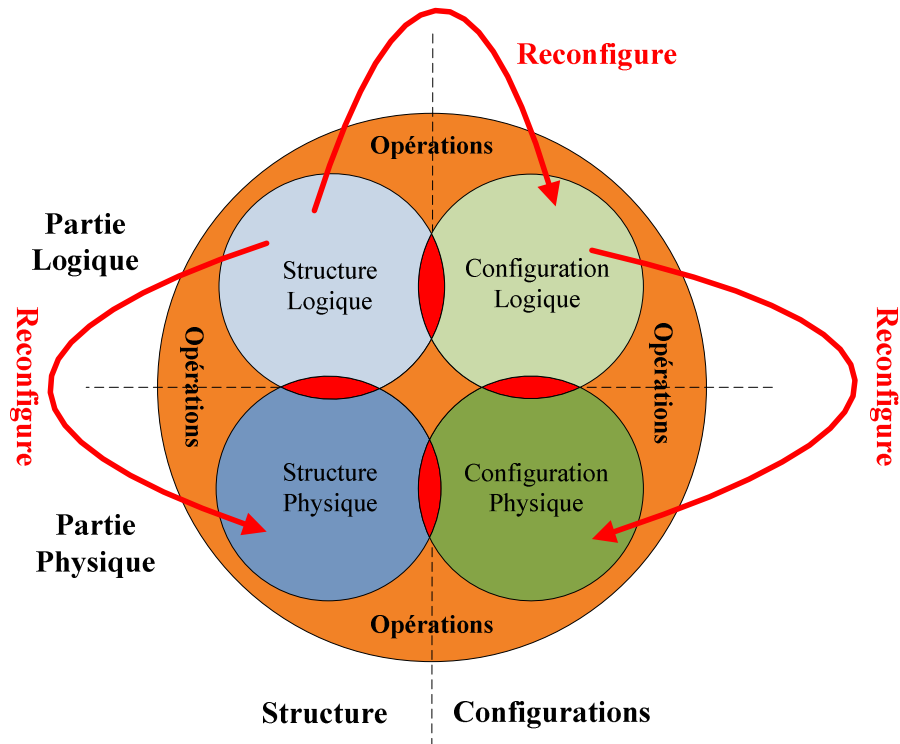


Figure 3.4. Intégration du processus de reconfiguration avec le modèle générique.

3. Modélisation des opérations de reconfiguration

Nous modélisons dans cette section les étapes nécessaires de reconfiguration sous forme de diagrammes de séquence et de diagrammes d'activité SysML qui sont capables de décrire le comportement du système.

3.1. Diagrammes d'activité

La modélisation des activités fait partie des principaux moyens pour décrire le comportement en SysML. Les diagrammes d'activité sont essentiellement concernés par le flux entre les actions: flux de contrôle et flux d'objet (les objets peuvent être des données, des éléments physiques tels que des pièces ou des fluides ou d'énergie).

Dans cette étude, le configurateur représente l'élément clé de notre système, il s'occupe particulièrement de rechercher et d'implanter les configurations. Nous distinguons donc les deux grandes missions du configurateur: la phase de recherche d'une nouvelle configuration, et la phase d'implantation de cette configuration. Nous construisons ainsi les diagrammes des activités SysML en utilisant Artisan Studio (**Atego, 2010**) concernant les deux cas d'utilisation principaux « Rechercher une configuration » et « Implanter une configuration » détaillés dans le 2^{ème} chapitre de la seconde partie.

Dans la phase de recherche d'une configuration, le configurateur choisit une configuration déjà conçue et enregistrée dans une bibliothèque de configurations s'il peut trouver une configuration qui peut atteindre l'objectif de la reconfiguration. Sinon, il conçoit une nouvelle configuration qui bien sûr enrichira la bibliothèque. Dans les deux cas, il doit enregistrer cette configuration (Figure 3.5).

En ce qui concerne la conception d'une nouvelle configuration, le configurateur doit remplir tous les paramètres définissant une configuration. Pratiquement, ces paramètres sont: le nombre de postes, le nombre de phases, le nombre de postes à chaque phase, les coordonnées cartésiennes de chaque ressource, les différentes connexions entre ces ressources en respectant les postes suivants à chacune, et les capacités demandées de chaque ressource. Ensuite, il ajoute cette nouvelle configuration à la base de données qui représente la bibliothèque (Figure 3.6).

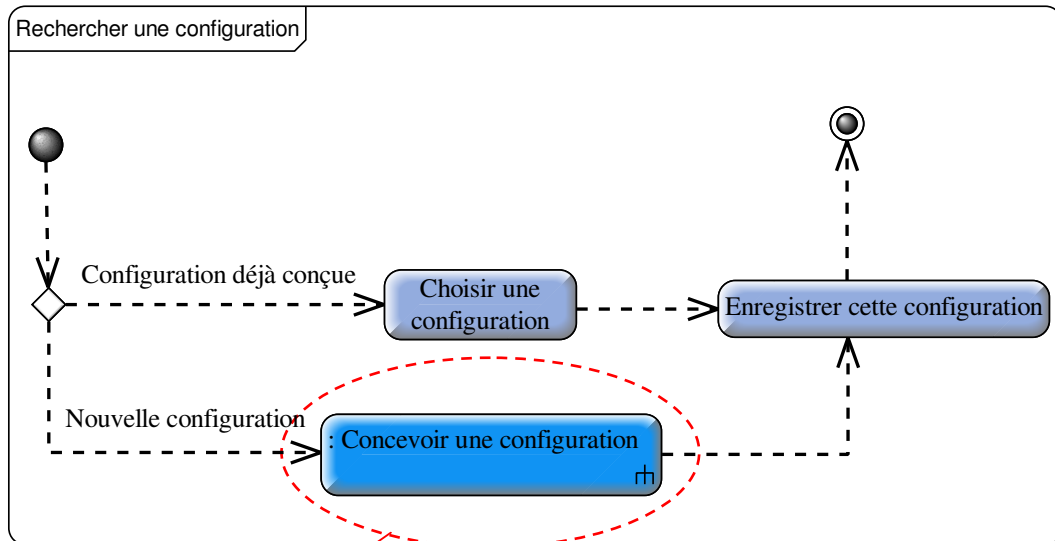


Figure 3.5. Diagramme d'activité illustrant la recherche d'une configuration.

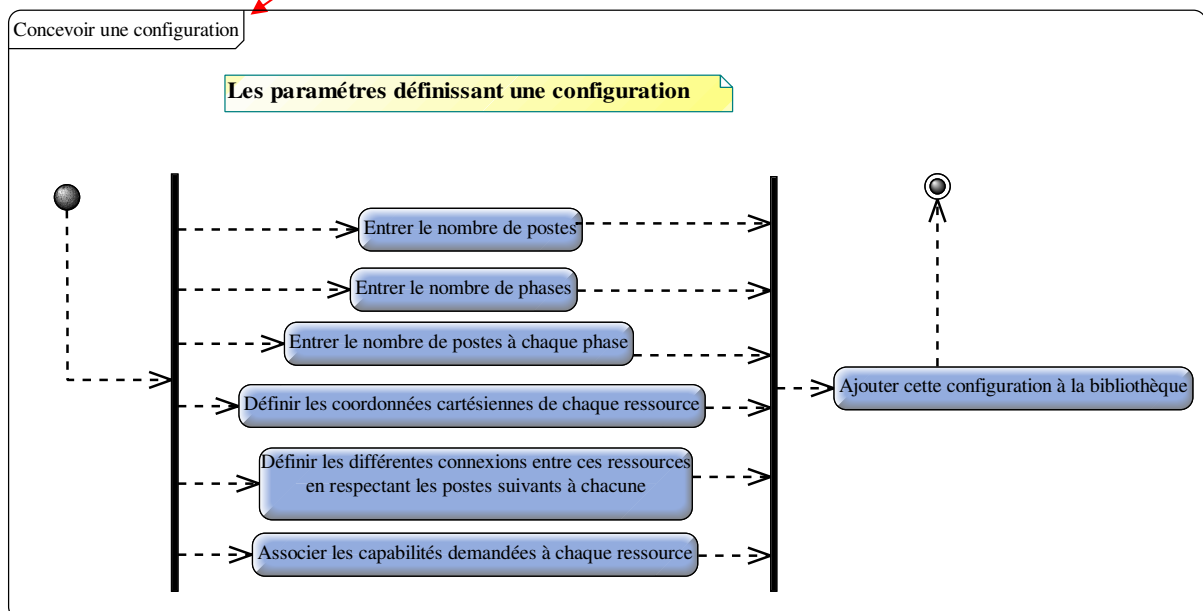


Figure 3.6. Diagramme d'activité illustrant la conception d'une configuration.

Au niveau de la phase d'implantation d'une configuration, le configurateur doit affecter la configuration, paramétrer de nouveau la commande et activer les modules nécessaires sur chaque ressource utilisée (Figure 3.7). Pratiquement, l'affectation d'une configuration conduit à affecter premièrement les ressources, puis arranger ces ressources, et enfin connecter ces ressources (Figure 3.8).

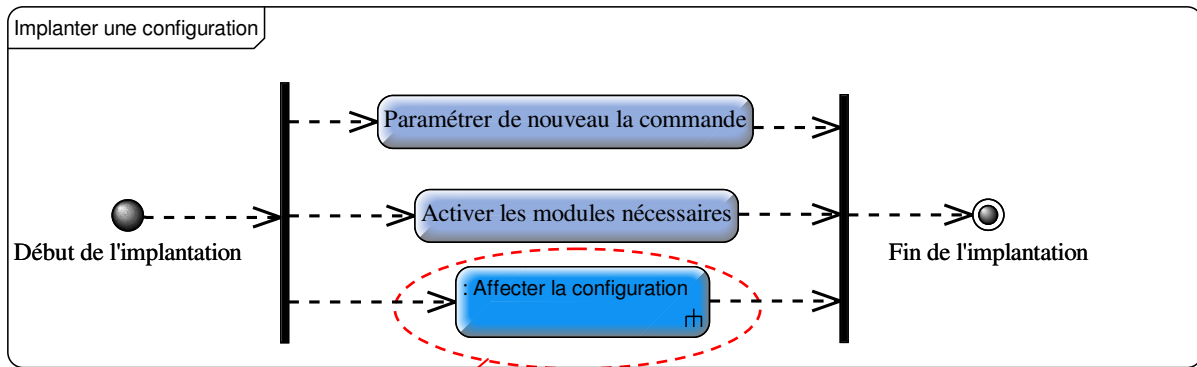


Figure 3.7. Diagramme d'activité illustrant l'implantation d'une configuration.

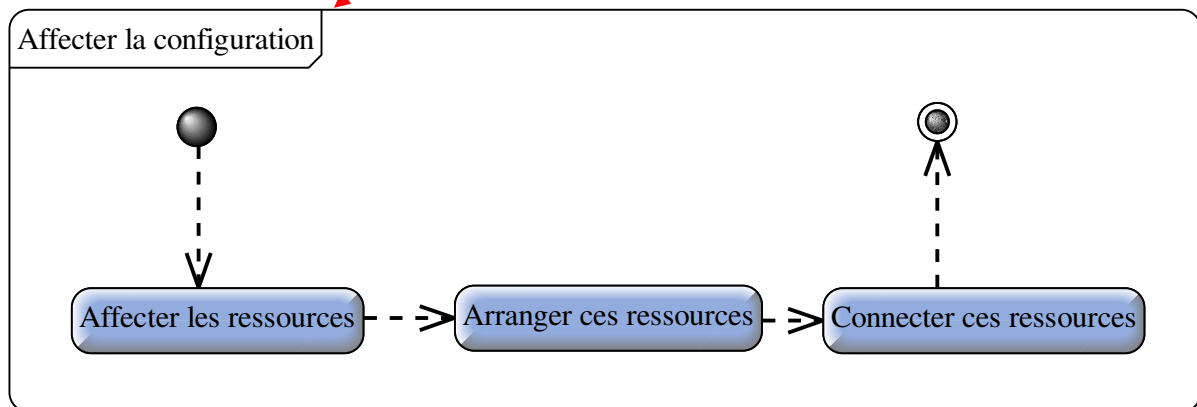


Figure 3.8. Diagramme d'activité illustrant l'affectation d'une configuration.

3.2. Diagrammes de séquence « système »

Le diagramme de séquence SysML illustre graphiquement la séquence verticale des messages passés entre éléments (lignes de vie) au sein d'une interaction. Nous considérons ainsi le système entier comme une boîte noire et le représentons par une seule ligne de vie. Le comportement du système est décrit vu de l'extérieur, sans montrer les interactions internes et comment nous les réalisons. En d'autres termes, il décrit chronologiquement les échanges de messages au sein d'un système.

Ce type de diagramme sert à mettre en évidence les échanges entre les systèmes et son environnement. Il sert donc à modéliser des séquences d'interactions entre une entité et son environnement extérieur, et aussi entre entités constituantes d'une sur-entité. En plus, il sert également à faciliter la réflexion sur le comportement dynamique des entités, et à la recherche de fonctions et d'interfaces (**Friedenthal et al., 2012**).

Dans ces travaux de thèse, nous nous intéressons à la fonction de reconfiguration, nous construisons donc les diagrammes de séquence SysML concernant les deux cas d'utilisation principaux « Rechercher une configuration » et « Implanter une configuration » en utilisant Artisan Studio (**Atego, 2010**).

En conséquence, nous illustrons deux diagrammes de séquence montrant les différentes interactions lors de la recherche d'une configuration, et également lors de l'implantation d'une configuration. Effectivement, ces interactions expliquent le rôle de chaque acteur et les fonctionnalités mises à sa disposition, ainsi que le rôle de chaque élément du système. Cela répondra à la question : « Qui fait quoi ? ».

Le premier diagramme montre l'interaction directe entre le configurateur, qui est le concepteur de notre système, et les éléments du système (une interface homme machine, une base de données), ainsi que la séquence des opérations à effectuer lors de la recherche d'une configuration (Figure 3.9).

Le deuxième diagramme illustre la séquence de messages échangés entre le configurateur et le système (acteurs, éléments structurels) lors de l'implantation d'une configuration (Figure 3.10).

Rechercher une configuration**Description****Premier cas: choisir une configuration déjà conçue de la bibliothèque**

Choisir une configuration

Récupérer cette configuration

Afficher la référence de cette configuration

Configuration est prête à implanter

Implanter cette configuration

Deuxième cas: concevoir une nouvelle configuration

Concevoir une nouvelle configuration

Transférer la demande

Tableau 1: caractériser la configuration

Entrer les premiers paramètres de cette configuration

Donner une référence à cette configuration

Transférer la référence à la base de données

Entrer le nombre de postes

Transférer le nombre de postes

Entrer le nombre de phases

Transférer le nombre de phases

Entrer le nombre de postes à chaque phase

Transférer le nombre de postes à chaque phase

Tableau 1 est complet

Tableau 1 est bien rempli et complet

Valider le tableau 1

Enregistrer tableau 1

Tableau 2: compléter la caractérisation de cette configuration

Entrer les paramètres concernant chaque poste

Activer les capacités de chaque poste

Transférer les capacités à la base de données

Définir une séquence des opérations selon la gamme logique

Transférer la séquence des opérations

Donner les coordonnées cartésiennes à chaque poste au niveau de l'atelier

Transférer les Coordonnées cartésiennes

Définir les postes suivants de chaque poste

Transférer les postes suivants de chaque poste

Tableau 2 est complet

Tableau 2 est bien rempli et complet

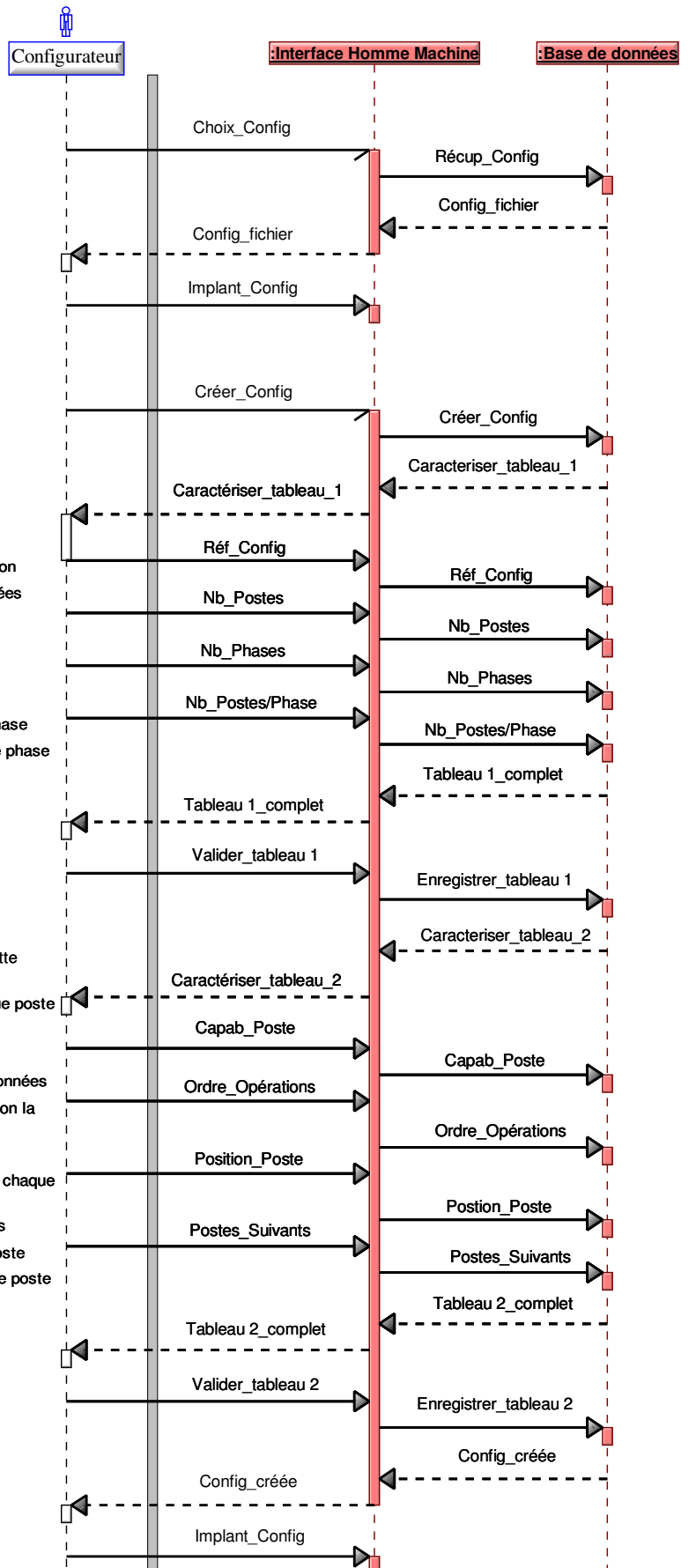
Valider le tableau 2

Enregistrer tableau 2

Configuration est bien enregistrée

Configuration est créée et prête à implanter

Implanter cette configuration

**Figure 3.9.** Diagramme de séquence lors de recherche d'une configuration.

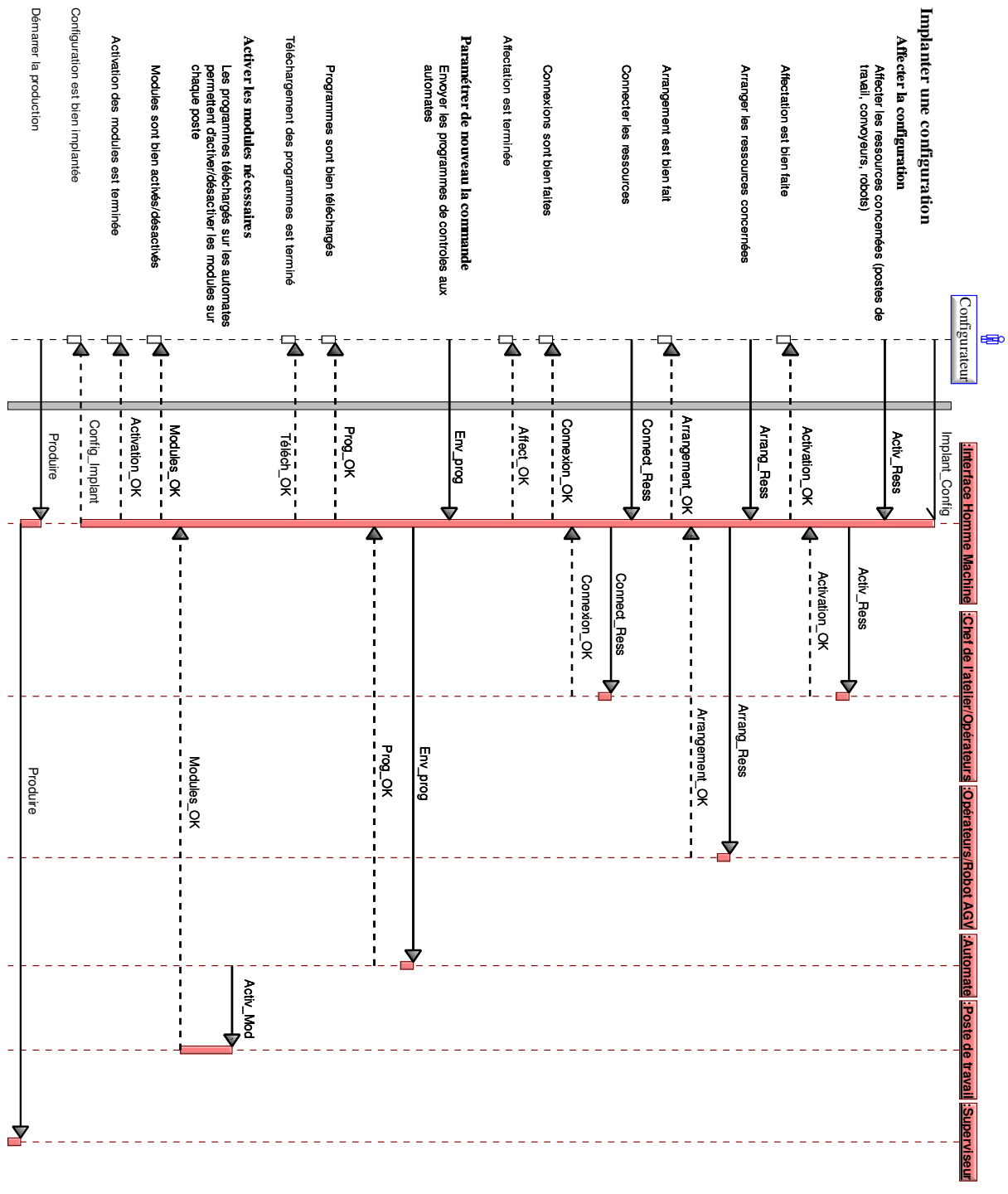


Figure 3.10. Diagramme de séquence lors d'implantation d'une configuration.

Par conséquent, nous avons ainsi proposé une méthodologie pour réaliser la fonction de reconfiguration. Un pilotage opérationnel compatible avec le SPRA sera présenté dans la suite de cette partie.

4. Architecture de Pilotage Hybride Adaptée au SPRA (APHA-SPRA)

Nous avons présenté toutes les architectures de pilotage dans le chapitre 4 de la première partie ainsi qu'une nouvelle Architecture de Pilotage Hybride Adaptée au SPRA (APHA-SPRA) peut être efficace et prometteuse pour répondre à l'objectif principal de ce type de système de production SPRA.

Dans notre cas, nous rappelons qu'il est composé de plusieurs phases et à son tour chaque phase est constituée de plusieurs postes de travail, et également d'un ordinateur de supervision.

Nous considérons qu'il existe un seul point décisionnel sur chaque poste de travail qui prend toutes les décisions concernant les flux informationnels et décisionnels de ce poste. Ce point central local gère en temps réel les événements et réalise la synchronisation et la coordination de toutes les tâches sur ce poste. Donc, il permet une meilleure gestion et une optimisation globale du fonctionnement au niveau du poste. Un pilotage centralisé peut alors être efficace sur chaque poste.

Bien sûr, nous avons besoin de coopérations entre les différents postes de travail. Un pilotage décentralisé est proposé au niveau de chaque phase afin de doter le SPRA d'une grande autonomie et différentes coopérations qui permettent une grande performance contre les perturbations et rendent le système facilement extensible.

Pour bien piloter le système d'une manière globale, il est nécessaire de réaliser la coopération entre les phases. Nous avons donc besoin de distribuer la décision centrale au niveau de SPRA sur un ensemble de centres de décision pilotant chacun une phase. Plus exactement, le pilotage est partagé entre plusieurs centres de décision dotés chacun de capacités d'autonomie et de coopération qui communiquent entre eux afin de mener à bien la production planifiée. Cela représente parfaitement un pilotage hétérarchique qui intègre les différents niveaux d'information et de décision dans l'atelier, et soutient fortement la réactivité par une meilleure intégration en temps réel des perturbations. Dans ce cas, l'ajout d'une entité ou la suppression temporaire pour cause de panne par exemple sont très faciles à mettre en œuvre, et cela est complètement compatible avec le système étudié SPRA.

De plus, le développement des systèmes contrôlés par le produit « intelligent » permet de créer des systèmes de production totalement réactifs car le produit « intelligent » est porteur d'informations et capable de faire varier son comportement en réponse aux différentes situations et exigences.

En conséquence, ce nouveau type de pilotage résulte d'un compromis entre les quatre approches : centralisée, décentralisée, hétérarchisée, et le pilotage par le produit. Au niveau de chaque poste, un pilotage centralisé est proposé tel qu'un

point local gère en temps réel toutes les tâches sur ce poste. En outre, pour chaque phase, nous proposons le pilotage décentralisé qui est capable de communiquer et de faire coopérer les postes pour satisfaire leurs objectifs locaux et résoudre les problèmes détectés. Au niveau de SPRA, le pilotage hétérarchique permet l'échange d'informations et la transmission de décisions en gardant ainsi la notion de supervision avec les différentes phases ainsi qu'au niveau des postes de travail. L'identification du produit « intelligent » va permettre de conduire le système en informant chaque poste de travail de ce qu'il faut faire. De même, ce produit peut être transformé et/ou transporté entre les composants du système.

Finalement, ce nouveau système de pilotage hybride a la possibilité d'évoluer entre un fonctionnement purement centralisé, un fonctionnement parfaitement décentralisé et un fonctionnement complètement hétérarchisé en bénéficiant des avantages du pilotage par le produit (Figure 3.11).

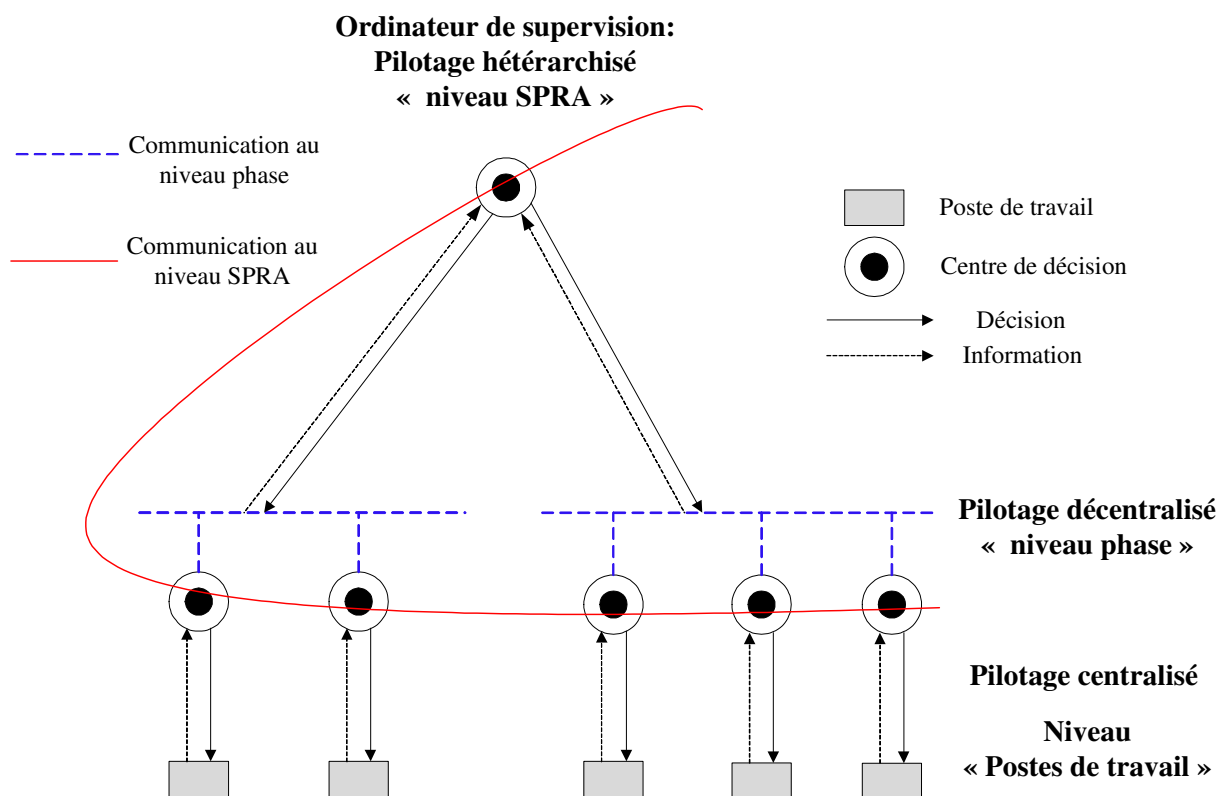


Figure 3.11. Proposition d'une nouvelle architecture de pilotage (APHA-SPRA).

Egalement, il est important de noter qu'un pilotage décentralisé basé sur le concept de systèmes multi-agents peut être prometteur dans notre cas de SPRA car il est capable de produire des solutions efficaces et de haute qualité en temps réel.

En ce qui concerne la commande, son architecture classique se décompose en quatre niveaux (Figure 3.12). Au niveau le plus bas, nous retrouvons le système physique et les entrées/sorties liées aux capteurs et aux actionneurs, ainsi que le

boîtier réalisant l'interface avec les produits. Le second niveau est constitué de nœuds de contrôle/commande, parmi ceux-ci se trouvent des automates programmables ainsi que des cartes électroniques qui peuvent se substituer aux automates programmables ou travailler avec eux. Le troisième niveau est composé d'un ordinateur chargé de la supervision du système ainsi que de l'interface avec l'utilisateur. Le niveau le plus élevé représente la planification globale de l'entreprise. De plus, une base de données est nécessaire pour mémoriser en détail toutes les informations, les ordres et le statut de l'atelier et des produits.

En outre, les liens entre les différents niveaux s'effectuent à travers des réseaux. Par exemple, la liaison entre les niveaux 1 et 2 peut être réalisée par un bus CANOpen alors que les équipements des niveaux 2 et 3 communiquent par le protocole Modbus TCP sur une liaison Ethernet.

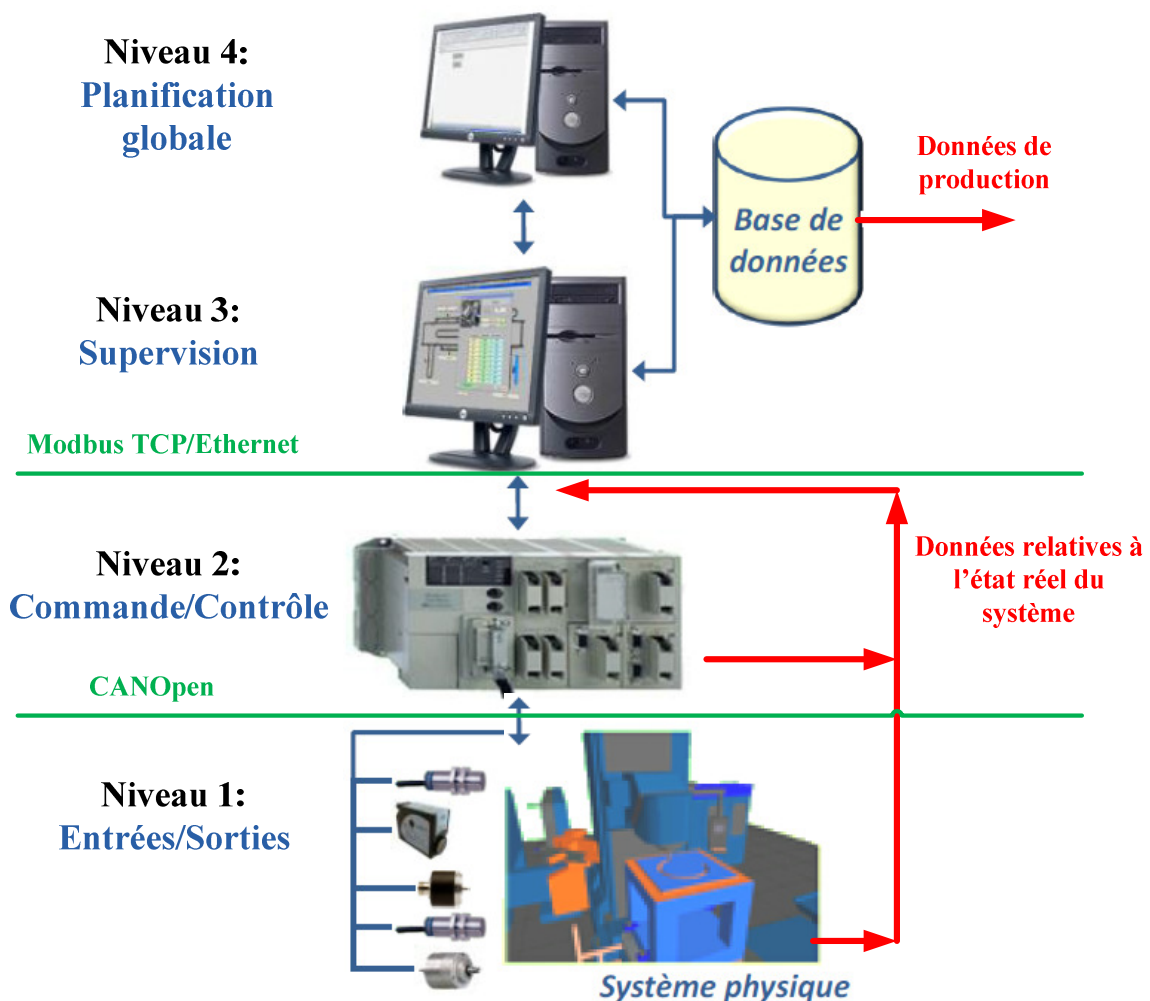


Figure 3.12. Architecture globale de commande.

Les travaux de thèse se situent entre les niveaux 2 et 3 (Figure 3.13) dans le cadre du développement d'un système de production reconfigurable et agile (SPRA).

Dans la section suivante, nous présentons nos apports proposant une nouvelle architecture de fonctionnement opérationnel compatible avec le SPRA.

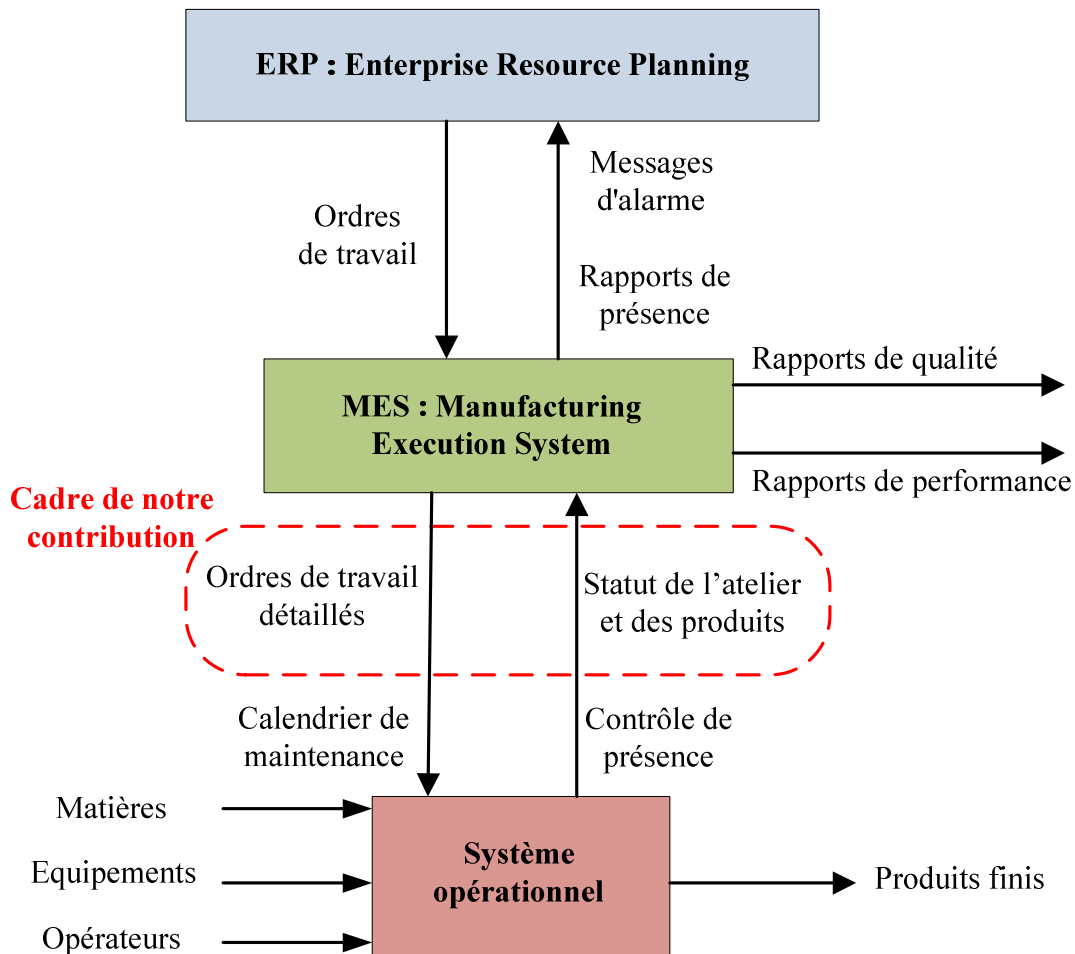


Figure 3.13. Principe général des différentes interactions dans l'entreprise.

5. Fonctionnement opérationnel du SPRA

Les systèmes de production et en particulier les systèmes de production reconfigurables et agiles (SPRA) peuvent être des systèmes complexes et difficiles à gérer au vu de toutes leurs composantes fonctionnelles (production, maintenance, gestion, pilotage,...) et également du développement continu et rapide de la technologie. Plusieurs approches ont été envisagées dans le but de mieux comprendre leur fonctionnement et de mieux les appréhender. D'un point de vue général, il est conventionnel de décomposer le système de production en trois sous-systèmes coopérants : le système physique, le système d'information et le système de décision (détaillés dans le premier chapitre de la première partie de ce mémoire).

En effet, dans un système de production, les sous-systèmes d'informations et de décision n'ont pas d'existence propre indépendamment l'un de l'autre. Ils

constituent ensemble ce que nous appelons le système de pilotage (**Trentesaux, 2002**), (**Tamani, 2008**). Ainsi, il est plus conforme à la réalité de considérer le système de production comme l'association d'un système physique et d'un système de pilotage (Figure 3.14).

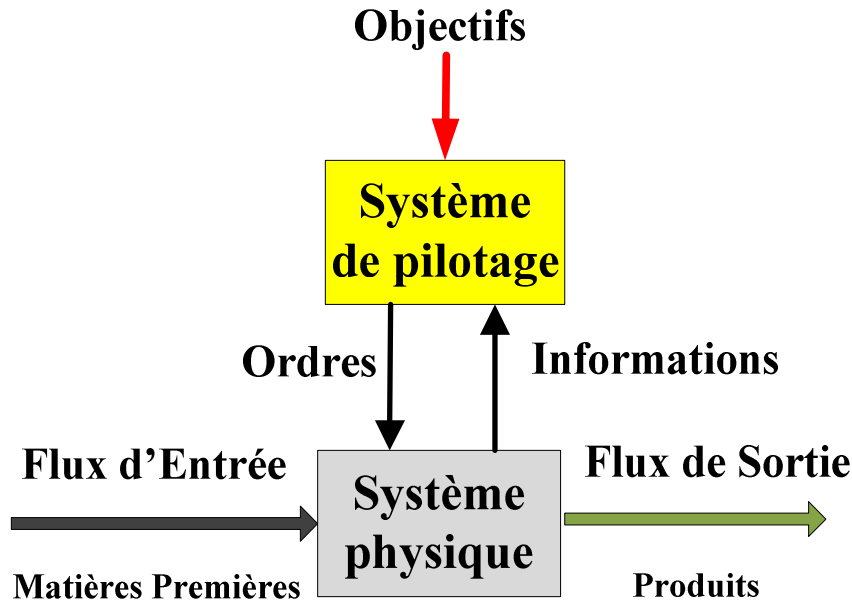


Figure 3.14. Décomposition d'un système de production en deux sous-systèmes.

Dans la littérature, il n'existe pas d'architecture de fonctionnement standard pour les systèmes de production. Par contre, plusieurs travaux de recherche ont déjà proposé des architectures de fonctionnement pour les systèmes de production habituels et peu pour les systèmes de production reconfigurables, parmi lesquelles nous pouvons citer les architectures du LAGIS (Laboratoire d'Automatique, Génie Informatique et Signal) (**Lee, 2006**), l'architecture du LAAS (Laboratoire d'Analyse et d'Architecture des Systèmes) (**Zamai, 1997**), l'architecture du LAG (Laboratoire d'Automatique de Grenoble) (**Mendez, 2002**), et également les approches du LESTER (Laboratoire d'Electronique des Systèmes Temps Réel) (**Mouchrad, 2002**) et (**Lamotte, 2006**).

Dans ces travaux de thèse, nous nous intéressons en SPRA qui est caractérisé par la modularité, l'intégrabilité, la convertibilité et la réactivité. Afin de bien piloter ce nouveau type de systèmes de production, son système de pilotage doit répondre à ces exigences et doit donc être conçue au départ de composants modulaires pour offrir des capacités de reconfiguration efficaces et rapides au niveau du procédé reconfigurable ainsi qu'au niveau de l'électronique de commande reconfigurable.

Nous proposons ainsi une nouvelle architecture de fonctionnement en prenant compte l'aspect de reconfiguration qui représente le cœur de cette architecture innovante (Figure 3.15). Généralement, nous pouvons distinguer deux niveaux

principaux de fonctionnement. Le niveau stratégique, le plus haut, décide quel produit est réalisé et comment il sera réalisé, et le niveau production qui se charge de la réalisation des produits. En ce qui concerne la production, nous plaçons la fonction de reconfiguration au centre du fonctionnement du système. Nous définissons ensuite les composants principaux de ce système de pilotage et nous expliquons les différentes interactions entre eux dans le but de réaliser un pilotage opérationnel pratique et efficace pour le SPRA. Le rôle de chaque composant est défini de la façon suivante.

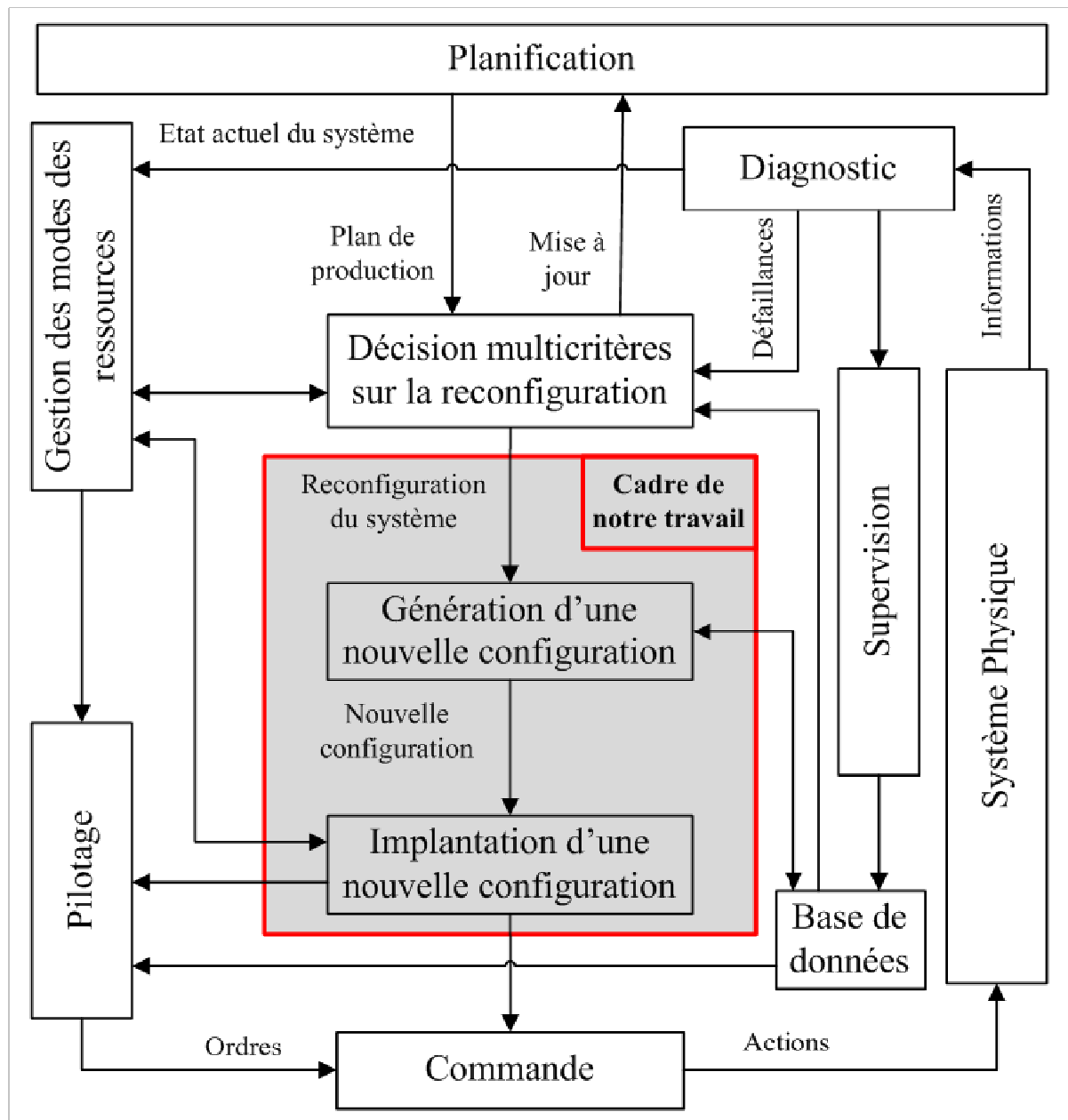


Figure 3.15. Proposition d'une architecture de fonctionnement opérationnel d'un SPRA.

Planification :

Ce composant représente la direction, la stratégie et la gestion de ressources de l'entreprise. Il contient toutes les applications permettant une vision à long terme de l'évolution du système. En outre, il consiste à définir un plan de production sur un horizon de temps en fonction du carnet de commande fourni par le client et des moyens de production. Ce plan fait apparaître des valeurs comme la quantité, la qualité, les délais de production et également une pré-allocation des ressources.

La planification conduit donc à planifier des reconfigurations qui interviennent dans le cadre du fonctionnement normal du système. Dans ce cas, la reconfiguration n'est pas considérée comme un traitement d'exception. Elle est néanmoins perçue comme une action modifiant le procédé.

Autrement dit, la reconfiguration peut être souhaitée par la planification lorsque l'objectif est atteint ou s'il y a besoin de lancer un nouveau produit.

Diagnostic de système :

Le diagnostic a pour objectif de suivre le fonctionnement du système, analyser ses performances et réagir dès l'apparition d'une défaillance ou d'une panne. Il s'agit donc de détecter et localiser les anomalies de moyens de production au niveau du système en utilisant des indicateurs de performances.

Le principe est de faire une comparaison entre le fonctionnement réel du procédé et son fonctionnement souhaité (cas idéal). Chaque indicateur ou sous ensemble d'indicateurs correspond à une panne bien particulière de manière à pouvoir localiser la défaillance après l'avoir détecté. Pratiquement, chaque indicateur ou sous ensemble d'indicateurs retourne une certaine valeur (zéro par exemple) qui signifie que le fonctionnement est normal au sein de sa partie, quand il retourne une autre valeur (un par exemple) nous avons donc un comportement défaillant au moins par un composant dans cette partie du système. Généralement, ils prennent la forme de capteurs. Ce non respect, pour la configuration courante, peut être entraîné par la défaillance d'un élément du système ou une modification de l'environnement. Il se traduit par des indicateurs qui dépendent de la grandeur mesurée ainsi que de la manière dont est mis en place le système.

Le résultat de ce composant est pris en compte par la décision pour déterminer si une reconfiguration est nécessaire.

Gestion des modes des ressources :

Généralement, la production et la maintenance sont les deux grandes familles de l'exploitation d'un système de production. Dans cette étude, nous ne traitons pas la famille « Maintenance ». Au niveau de la famille « Production », nous définissons quatre modes « Fonctionnement », « Production », « Marche » et « Arrêt ». Chaque mode est un ensemble d'états caractérisant une ressource ou un ensemble de ressources selon un point de vue.

L'évolution des modes des différentes ressources nécessite un mécanisme garantissant une cohérence dans cette évolution. Ce mécanisme est appelé la gestion des modes des ressources qui joue plusieurs rôles. Il donne premièrement une information sur l'utilisation des différentes ressources, il aide à l'évaluation du coût de la reconfiguration et il a la charge de faire respecter les contraintes associées aux modes des ressources et de leurs opérations pendant la reconfiguration.

Décision multicritères sur la reconfiguration :

Ce composant dirige le processus de reconfiguration décrit dans la deuxième section de ce chapitre.

Effectivement, plusieurs critères peuvent guider le configurateur à prendre sa décision importante sur la reconfiguration du système. Des critères pris en compte sont essentiellement les besoins du client, le temps nécessaire à la reconfiguration, la complexité de sa mise en œuvre, son coût de mise en place, ses performances et le coût de fonctionnement de la nouvelle configuration. Egalement, d'autres critères doivent être pris en compte comme le résultat du diagnostic et l'état actuel du système dans la configuration actuelle : état des produits, mode des ressources.

Génération d'une nouvelle configuration :

Nous définissons ce composant clé qui rend notre proposition innovante. Ce composant tient une place importante dans le fonctionnement du système tel qu'il détermine les moyens mis en œuvre pour réaliser un objectif ainsi que les opérations demandées par la commande permettant de répondre aux besoins du client.

L'objectif de ce composant est de générer une nouvelle configuration. Le choix/la construction d'une nouvelle configuration constitue une étape centrale de notre travail. L'approche utilisée est déjà proposée et explicitée dans la section « Processus de reconfiguration » de ce chapitre, ce processus est intégré avec le modèle générique proposé dans le premier chapitre de la deuxième partie de ce mémoire (voir Figure 3.3 et Figure 3.4).

Cela peut être utilisé conjointement : nous pouvons alors utiliser une bibliothèque de configurations déjà conçues dans laquelle une nouvelle configuration est choisie. Comme tous les cas de pannes ne peuvent pas nécessairement être identifiés avant le fonctionnement, si aucune configuration n'est compatible avec l'état du système, la construction d'une nouvelle configuration peut être envisagée. Cette configuration vient alors enrichir la bibliothèque et permettra une réaction plus rapide face à un nouvel aléa similaire.

Cette génération des configurations peut être faite en ligne ou hors ligne. Une analyse hors ligne permet de classer les différentes configurations pour répondre à diverses utilisations du système, une analyse en ligne peut la compléter en prenant en considération le coût de reconfiguration qui dépend de l'état courant du système.

Pratiquement, le choix/la construction d'une nouvelle configuration nécessite d'utiliser des techniques très dépendantes de l'application. Plusieurs méthodes peuvent être envisagées :



Une méthode qui consiste à réorganiser la configuration courante ;



Une méthode par construction est aussi possible, il s'agit de composer différentes configurations en utilisant des configurations minimales déjà définies, cette technique met à la fois en œuvre un choix et une construction de la configuration.

Implantation d'une nouvelle configuration :

Nous définissons le second composant clé qui suit le résultat du composant précédent et complète ainsi la fonction de reconfiguration.

Ce composant est consacré à l'implantation d'une nouvelle configuration qui fait éventuellement intervenir le pilotage et la gestion des modes afin de mettre le système et les produits dans un état compatible avec la nouvelle configuration.

Dans cette phase, le configurateur doit affecter la configuration, paramétrer de nouveau la commande et activer les modules nécessaires sur chaque ressource utilisée. Pratiquement, l'affectation d'une configuration conduit à affecter premièrement les ressources, puis arranger ces ressources, et enfin connecter ces ressources.

D'une manière générale, lors du passage d'une configuration A à une autre configuration B, nous implantons de nouveau les programmes de contrôle sur chaque ressource utilisée dans cette configuration. Par exemple, au sein d'une gamme logique, nous activons les opérations demandées d'un poste et en même temps nous désactivons les autres opérations que ce poste est capable de faire.

En ce qui concerne l'activation des modules, il s'agit d'activer des outils et/ou dispositifs sur les ressources utilisées selon les exigences de la configuration. Par exemple, nous branchons une visseuse et en même temps nous débranchons une perceuse dans un poste prévu à ces effets.

Supervision :

La supervision et le pilotage sont deux fonctions du système décisionnel dont les rôles sont inverses mais complémentaires.

Le rôle de la supervision consiste à collecter les données, traiter les informations, alimenter les bases de données, diffuser les résultats, en construisant ainsi des jeux d'informations représentant l'état actuel du système opérant. En d'autres termes, la supervision a pour objectif de contrôler l'exécution d'une opération ou d'un travail effectué sans rentrer dans les détails de cette exécution, elle recouvre ainsi l'aspect fonctionnement normal et anormal.

Pilotage :

Le pilotage a pour objectif de donner les ordres de fonctionnement au système de production, il doit donc indiquer de manière très concrète les actions à conduire à court terme ou bien en temps réel pour obtenir un fonctionnement efficace et efficient du système de production.

Au niveau de la reconfiguration, le pilotage est responsable de l'affectation d'une configuration. Pour cela, il prend en compte les informations disponibles sur l'état et l'utilisation des différentes ressources, et aussi les informations stockées dans la base de données, ces informations représentent effectivement l'état actuel du système opérant. Ensuite, il donne les ordres utiles pour affecter, arranger et connecter les ressources utilisées dans cette configuration. Egalement, ce composant est chargé de paramétrer de nouveau la commande et d'activer les modules nécessaires sur chaque ressource utilisée.

Ce composant permet alors d'indiquer qui fait quoi, quand, où, et comment, en respectant d'une part les contraintes définies au niveau décisionnel supérieur (implantation d'une nouvelle configuration) et d'autre part celles inhérentes au système de production et à son environnement (base de données et gestion des modes des ressources).

Commande :

La commande est le niveau le plus bas de la structure décisionnelle. Elle expose au pilotage les différentes opérations qui peuvent être réalisées sur le système physique et elle réalise ainsi l'interface avec le procédé.

Dans un système reconfigurable, la commande est relative à la configuration courante. En effet, les opérations qui peuvent être réalisées sur le système sont définies pour cette configuration, les actions envoyées aux ressources ne peuvent donc pas sortir du cadre défini par cette configuration.

Par conséquent, nous avons présenté les différents composants permettant de mettre en œuvre un système de pilotage adapté au SPRA.

6. Conclusion

Dans cette partie, une méthodologie pour réaliser la fonction de reconfiguration a été proposée. Premièrement, nous avons proposé un processus de reconfiguration qui nous permet de rechercher une nouvelle configuration, et également de la mettre en place pour répondre aux nouveaux besoins de marché et/ou d'assurer une continuité du service avec une réorganisation du système. De plus, une modélisation des opérations de reconfiguration a été proposée sous forme de diagrammes des séquences et de diagrammes des activités SysML.

Egalement, nous avons proposé une nouvelle architecture de pilotage (APHA-SPRA) et un pilotage opérationnel compatible avec le SPRA en présentant une

architecture globale de commande et en expliquant les différentes interactions entre les composants principaux du système de pilotage afin de réaliser un pilotage opérationnel pratique et efficace pour un système de production reconfigurable et agile (SPRA).

Dans la dernière partie de ce mémoire, nous allons donc projeter les travaux et méthodes développés au cours de cette thèse sur un démonstrateur industriel dans le cadre d'une application du laboratoire.

Quatrième Partie

Application à une ligne d'assemblage de type SPRA

Cette dernière partie de ce mémoire est consacrée à l'application de nos recherches à un cas particulier d'un Système de Production Reconfigurable et Agile (SPRA). Nous présentons les expérimentations réalisées sur la simulation d'une ligne d'assemblage de cartes électroniques installée au sein de l'atelier AIP-PRIMECA Auvergne à l'IFMA.

Cette application développée dans le cadre du projet Agil@Prod de l'Institut Pascal de Clermont-Ferrand (UMR 6602 CNRS/UBP/IFMA) a pour but de montrer l'applicabilité des différents concepts étudiés dans cette thèse.

Le projet Agil@Prod a été financé par le programme de recherche du Gouvernement Français "Investissements d'avenir" à travers le Laboratoire d'Excellence IMoBS3 (ANR-10-LABX-16-01), par l'Union Européenne à travers le programme de compétitivité régionale et emploi 2007-2013 (Fonds FEDER – Région Auvergne) et par le conseil régional d'Auvergne.

Sommaire

1. Introduction	179
2. Projet Agil@Prod	179
3. Présentation des ressources utilisées	181
3.1. Poste de travail	182
3.2. Robot mobile AGV	184
4. Cas d'étude « assemblage des cartes électroniques »	185
5. Projection de nos approches	191
6. Interfaces de reconfiguration	204
7. Conclusion	208

1. Introduction

Afin de montrer l'intérêt des approches proposées dans ce mémoire et également d'immerger les développements théoriques dans un contexte réel, nous avons choisi un cas d'étude qui concerne l'assemblage de cartes électroniques. En effet, celui-ci présente régulièrement des besoins en reconfiguration de façon rapide et efficace. Autrement dit, cette partie a pour objectif de montrer l'applicabilité des approches proposées.

Tout d'abord, nous présentons le projet Agil@Prod dans lequel cette thèse s'inscrit. Puis, nous introduisons les cartes électroniques d'une manière générale. Nous nous intéressons uniquement à la partie montage des composants et nous présentons la ligne d'assemblage d'une manière fonctionnelle et organique.

Ensuite, nous projetons le modèle générique proposé dans ce mémoire sur cette ligne en distinguant ses quatre parties (structure physique, structure logique, configuration physique, configuration logique) et les six types d'opérations. Nous appliquons la méthodologie proposée qui définit les étapes à suivre pour concevoir et implanter une nouvelle configuration. Finalement, nous intégrons ainsi le processus de reconfiguration avec le modèle générique.

Enfin, nous présentons quelques manipulations qui servent à reconfigurer la ligne et déployer une configuration. D'abord, nous créons les interfaces nécessaires qui permettent au configurateur et au chef de l'atelier de réaliser quelques fonctions. Puis, nous proposons une base de données qui permet de gérer toutes les configurations de l'atelier. De plus, nous proposons des programmes automates et les interfaces homme machine (IHM) associés.

2. Projet Agil@Prod

Pour les entreprises industrielles, construire des systèmes de production capables de s'adapter rapidement aux différentes modifications de leur environnement, est un atout essentiel pour leur compétitivité et leur rentabilité.

Dans ce contexte, le projet Agil@Prod a été développé d'une manière conjointe entre l'Institut Français de Mécanique Avancée (IFMA) et le laboratoire de recherche Institut Pascal (IP) au sein de la plateforme AIP-PRIMECA Auvergne. La plateforme AIP-PRIMECA est une structure de mutualisation de moyens et d'expertise pour les besoins de la formation et de la recherche.

Plusieurs financements ont contribué à l'acquisition des équipements du projet Agil@Prod dont les plus importants sont :

- Le Contrat de projet Plan Etat-Région (CPER) Innov@pôle ;
- Le programme « Investissement d'Avenir » à travers le Laboratoire d'Excellence IMoBS3 ;

- Les fonds propre de l'établissement d'accueil l'IFMA (Figure 4.1) ;
- Le Gouvernement Syrien à travers l'Institut Supérieur de Sciences Appliquées et de Technologie (ISSAT) qui a financé la bourse de cette thèse.



Figure 4.1. Atelier AIP-PRIMECA Auvergne de l'IFMA.

Le but de ce projet est de contribuer au développement d'un concept innovant de lignes de production reconfigurables et agiles. Il s'intéresse particulièrement au développement des parties pilotage, commande et système d'information qui devront permettre l'agilité de la partie physique composée de plusieurs postes de travail reconfigurables et déplaçables soit manuellement soit à l'aide d'un robot mobile.

Parmi les questions auxquelles le projet a essayé de répondre, on peut citer :

- Comment gérer la reconfiguration?
- Comment assurer la communication et la coordination entre les différents modules ?
- Comment générer automatiquement des modules de commande pour s'adapter aux transformations de la partie matérielle ?
- Quelle répartition de l'intelligence entre le pilotage central, les postes de travail et les produits ?
- Quelle interfaçage entre la partie planification et la partie contrôle d'exécution ?

- Quand reconfigurer?
- Vers quelle configuration?

Pour mener à bien ce projet, nous avons préconisé une méthode qui s'appuie sur une approche par composants. Le système a été, ainsi, construit à base de composants matériels et logiciels qui collaborent pour former un ensemble cohérent auto-organisé. Doté de ces caractéristiques, le système est capable de se reconfigurer au niveau physique et au niveau logique pour chaque lancement d'un nouveau produit de façon à optimiser la production et à rationaliser l'utilisation des ressources.

Outre les apports d'Agil@Prod pour les aspects recherche, il a été conçu pour intéresser également les acteurs industriels. Le schéma ci-dessous présente une ébauche des différents cas d'utilisation envisagés (Figure 4.2) :

Pour l'industrie :

Cette plate-forme permet de tester des configurations, d'intégrer des nouveaux concepts, et de valider des nouvelles approches.

Pour la recherche :

La mise à disposition de ressources pour les équipes de recherche de laboratoires publics et privés afin de développer de nouvelles approches des systèmes de productions.

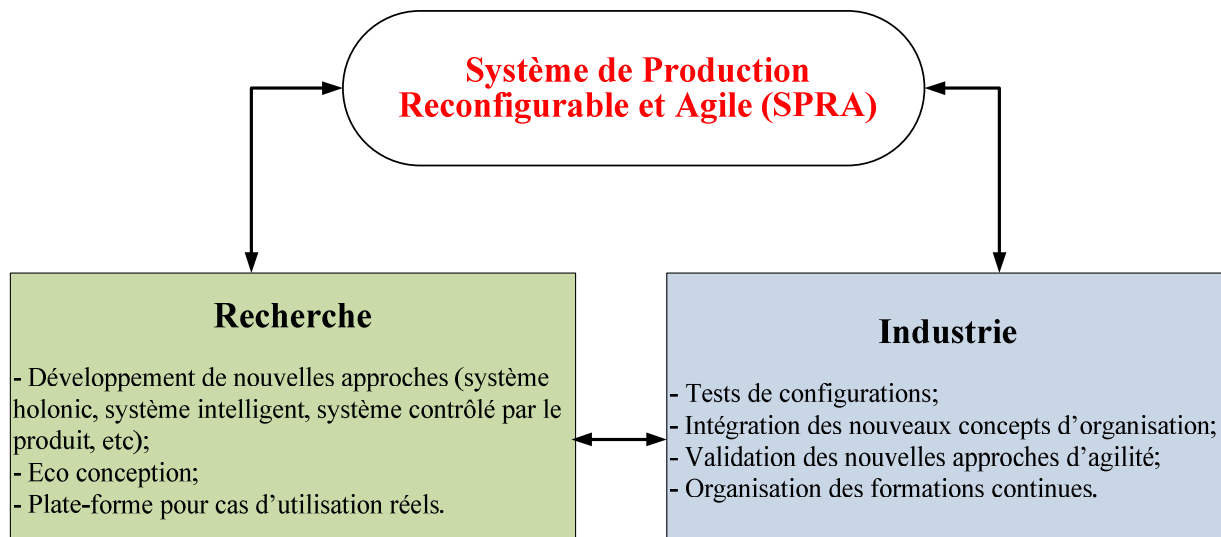


Figure 4.2. Dimensions industrielles et pédagogiques du projet Agil@Prod.

3. Présentation des ressources utilisées

Dans le cadre du projet Agil@Prod, nous avons défini fonctionnellement et organisationnellement les postes de travail, le robot mobile AGV et leur environnement.

3.1. Poste de travail

L'organisation des postes de travail est soumise à plusieurs contraintes du fait de leur mobilité. Mais ces postes doivent aussi faciliter le travail de l'opérateur et être ergonomiques. Chaque poste répond aux besoins fonctionnels suivants :

- Une interface homme machine qui permet la traçabilité ainsi que la supervision ;
- La capacité de renseigner sur l'état du produit ;
- Les commandes de base : comme des interrupteurs permettant la communication avec l'opérateur, bouton poussoir d'arrêt d'urgence,...
- L'habilité à être transporté.

Par conséquent, ces besoins fonctionnels imposent les composants suivants : Automate, écran, capteurs, moteur, convoyeur, alimentation, et boîtier de commande qui possède le bouton d'arrêt d'urgence et de mise en marche du convoyeur, celui-ci doit être par conséquent accessible facilement à l'opérateur.

Pour supporter la partie commande et l'interface opérateur, nous avons choisi le matériel Pro-face qui a l'avantage d'associer dans un même composant un automate et un écran tactile (**Proface, 2013**). Ce qui est sur le plan économique plus intéressant qu'un automate allié à un pupitre opérateur. La Figure 4.3 représente l'écran tactile Pro-face et un module d'Entrées/Sorties qui ont été utilisés pour les postes de travail. La programmation de la commande et la création des interfaces opérateurs se font avec un environnement de programmation spécifique GP-pro EX2.6 comme nous allons voir dans la suite de cette partie.



Figure 4.3. Automate et écran tactile Pro-face.

En outre, chaque poste de travail est équipé d'un convoyeur qui sert à transporter les pièces de l'entrée à la sortie de chaque poste et permet également aux postes de se mettre l'un à côté de l'autre pour faire un convoyeur global qui sert donc à

transporter les pièces entre les postes. Nous pouvons ainsi reconfigurer la partie transitique.

Par ailleurs, l'automatisation des postes demande obligatoirement la présence de capteurs pour récupérer les informations nécessaires à l'automate. Pour que le poste fonctionne, il faut que le convoyeur puisse s'arrêter afin que l'opérateur se saisisse de la pièce. Puis, l'identification de la pièce est nécessaire pour avoir une traçabilité et un contrôle sur la production étendue. Autrement dit, le poste doit pouvoir détecter la présence d'une pièce et l'identifier.

Pour remplir la fonction d'identification de la pièce, la technologie radio-identification nous a paru adaptée et accessible. La radio-identification plus souvent désignée par le sigle RFID (Radio Frequency IDentification) est une méthode pour récupérer et mémoriser des données à distance. Elle repose sur des étiquettes appelées aussi « radio-étiquettes » ou « RFID tag » ou « RFID transponder » qui sont collées sur les objets à identifier. Ces étiquettes contiennent quelques dizaines d'octets de mémoire. L'autre élément indispensable de cette technologie est le lecteur RFID qui, grâce aux ondes radio courte portée, peut lire les données des étiquettes. Le principe de cette technologie s'illustre grâce au schéma suivant (Figure 4.4). Nous avons, ensuite, envisagé de coupler ce lecteur RFID avec un capteur de proximité optique pour situer précisément les pièces parce que le lecteur RFID possède un champ d'interrogation variant d'un modèle de lecteur à un autre et le transfert de données s'effectue dans cette zone.

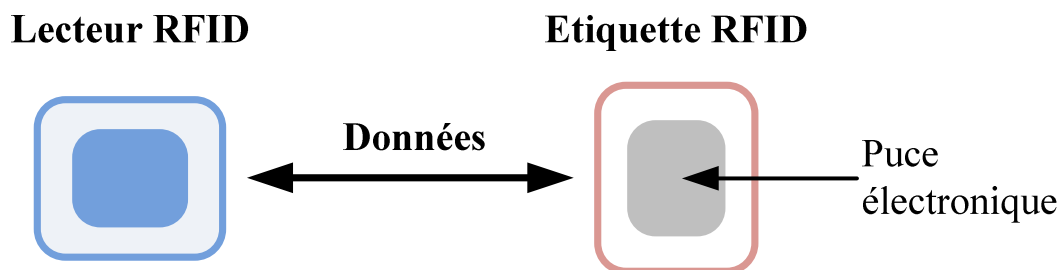


Figure 4.4. Principe de la technologie RFID.

Enfin, voici une photo d'ensemble de différents composants du poste de travail (Figure 4.5) :

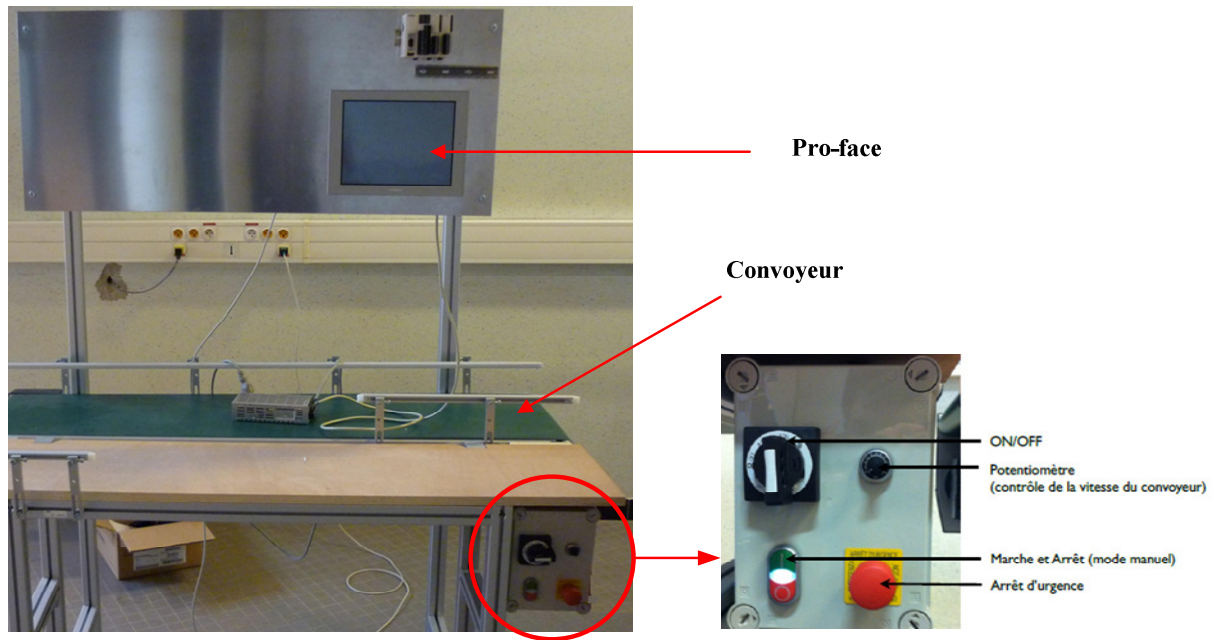


Figure 4.5. Poste de travail.

3.2. Robot mobile AGV

Tout d'abord, rappelons qu'un robot mobile sert pour le transport des pièces en entrée, en sortie de la ligne d'assemblage et aussi entre les postes si c'est nécessaire. L'AGV avec son convoyeur qui s'interface avec les postes de travail peut être reconfiguré pour obtenir des structures de convoyeurs avec des morphologies différentes. Cette idée est le concept de base d'Agil@Prod.

Le robot que l'on utilise a été conçu à l'IFMA et réalisé par la société BA Systèmes à Mordelles (Figure 4.6). Il s'agit d'un véhicule qui dispose de trois roues. Une roue peut être commandée en orientation et en vitesse de rotation tandis que les deux autres sont fixes en orientation et libres en rotation. Pour se repérer dans l'atelier, il dispose d'un codeur quantifiant la rotation de la roue motrice (qui donne donc une position relative) et d'un scrutateur laser pour calculer sa position absolue dans l'atelier. Pour commander son déplacement, on ne modifie que deux variables qui sont l'angle de braquage et la vitesse de rotation de la roue. Ce robot est aussi équipé d'un convoyeur qui se met en continuité les convoyeurs des postes de travail. Il dispose également d'une fonction qui lui permet de soulever un poste de travail si c'est nécessaire. La commande du robot est portée par un automate Pro-face avec une interface tactile permettant d'accéder à certains paramètres du robot.

Egalement, un lecteur RFID est présent sur ce robot pour renseigner sur le poste de travail qu'il va transporter ou pour lui donner un ordre de mission. Il suffit de passer la bonne carte sur le lecteur pour qu'il lise son numéro qui peut être associé par la suite à une mission. Un émetteur/récepteur WiFi lui permet d'échanger les données avec l'extérieur.

Pour la sécurité, le robot est équipé d'un système de contrôle de zone (plan) avec deux types de champs (alarme et protection) paramétrables, d'arrêts d'urgence et de bordures sensibles.

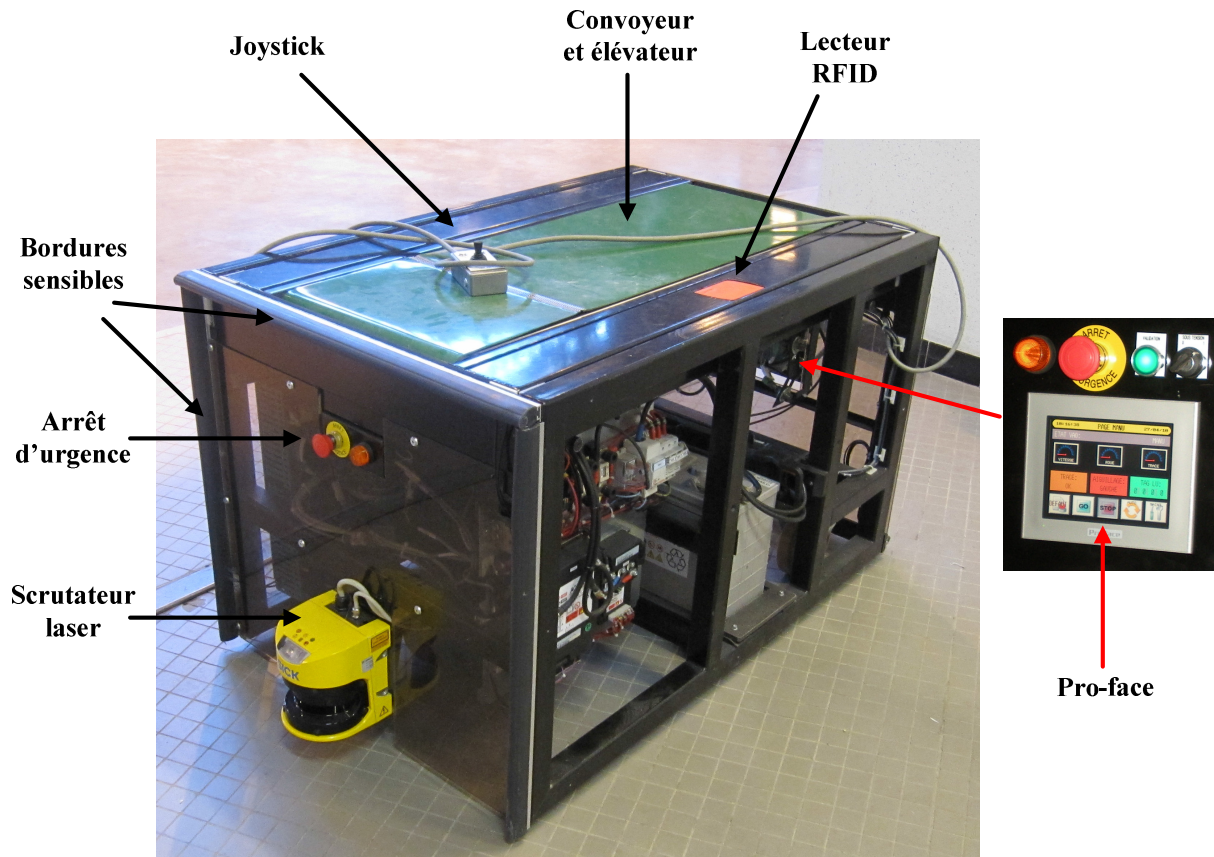


Figure 4.6. Robot mobile AGV.

Dans la suite de cette partie, nous projetons quelques uns de nos approches proposées dans ce mémoire de thèse sur le processus d'assemblage de cartes électroniques.

4. Cas d'étude « assemblage des cartes électroniques »

Dans cette section, nous montrons le besoin du concept SPRA dans les entreprises. La validation du modèle générique et de la méthodologie de conception d'un SPRA sera présentée et explicitée dans la section suivante.

Nous avons choisi un cas d'étude qui concerne l'assemblage de cartes électroniques car il présente régulièrement des besoins en reconfiguration d'une façon rapide et efficace.

Les cartes électroniques sont présentes dans de nombreux composants de notre quotidien : dans l'informatique notamment (carte graphique, box ADSL, carte

mère,...), dans les applications automatiques et temps réel (automobile, aéronautique), ou encore par exemple dans les boîtiers autonomes d'éclairage de sécurité.

Une carte électronique est composée des éléments suivants (Figure 4.7) :

- Un circuit imprimé : composé de plusieurs couches de cuivre séparées par un matériau isolant, il est généralement recouvert d'une couche de vernis. Dessus, sont gravées des pistes permettant la liaison électrique entre les composants. En plus, certains composants sont directement montés en surface (Composants Montés en Surface, ou CMS). Enfin, le circuit est percé en différents endroits afin de permettre l'insertion des composants.
- Des composants électroniques : il en existe de très nombreux types (Circuits Intégrés, transistors, condensateurs,...). Ils sont insérés et soudés dans ou sur le circuit imprimé.



Figure 4.7. Exemple de carte électronique.

Dans le cadre de cette thèse, nous nous intéressons uniquement à la partie montage des composants, et non à la fabrication des circuits imprimés. Les circuits imprimés sont souvent reçus prêts à l'emploi par les monteurs de composants.

Classiquement, le circuit imprimé passe dans un robot dit « robot CMS », qui est chargé de monter à une cadence extrêmement rapide les très petits composants, qu'ils soient CMS ou pas. Après cette opération, plusieurs postes manuels sont chargés de placer les composants plus gros (exemple : port S-véo sur une carte

graphique). Cette opération demande une dextérité importante et est réalisée par des ouvriers qualifiés. Puis le circuit imprimé et ses composants passent sur une vague d'étain liquide à plus de 232°C, qui permet de souder les composants. Un contrôle visuel par comparaison avec une carte «étalon» est effectué. Enfin, un opérateur place la carte finie dans une baie automatique de contrôle qui vérifie l'ensemble de ses fonctionnalités en quelques minutes.

Nous constatons que seules les opérations manuelles vont varier (nombre et type), alors que le passage par le robot, la vague d'étain et le contrôle seront toujours les mêmes (aux réglages près).

Dans la suite de ce paragraphe, nous présentons un exemple montrant le besoin d'intégrer les notions de la reconfigurabilité et de l'agilité dans les entreprises et surtout sur les lignes d'assemblage. Nous rappelons que nous nous intéressons uniquement à la partie montage des composants.

Prenons pour exemple le cas d'une entreprise qui réalise l'assemblage de cartes électroniques, alliant CMS et composants à monter à la main. Cette entreprise possède deux lignes d'assemblage, chacune étant dédiée à un type de cartes électroniques : la ligne 1 pour les cartes de type A et la ligne 2 pour les cartes de type B. Les deux gammes d'assemblage utilisent les mêmes technologies, les mêmes types de postes manuels, et ont certains composants en commun. En revanche, les opérations manuelles n'interviennent pas avec les mêmes types de composants, et ne sont pas en nombre égal. Chaque poste de montage manuel est spécifique à sa fonction (exemple : poste 1 monte uniquement les composants Comp1, Comp2 et Comp3). En effet, afin d'éviter toute inversion de composants, seuls ceux qui sont à monter sont présents et dans l'ordre de montage. Les gammes d'assemblage, qui se composent de différentes opérations, sont présentées dans le tableau suivant (Tableau 4.1) :

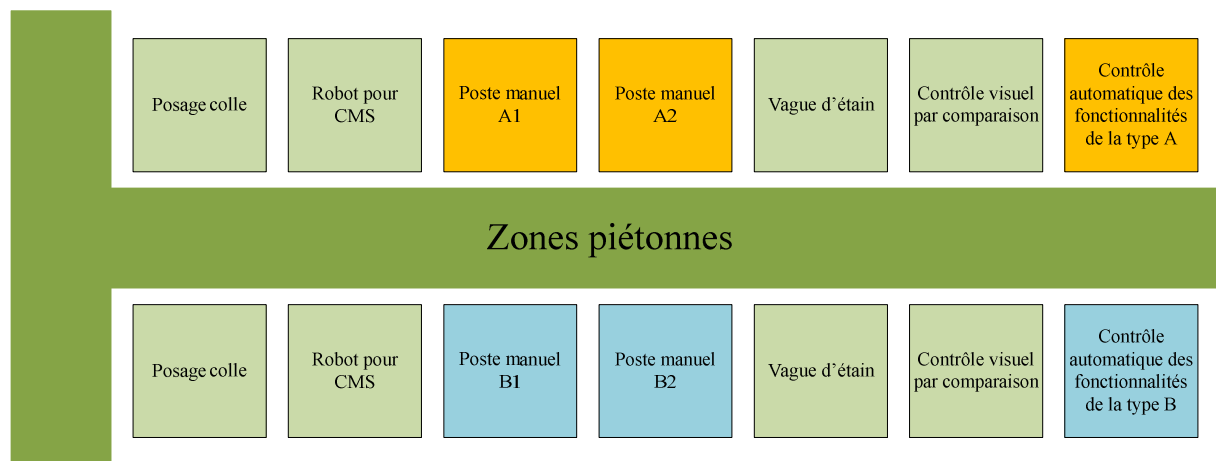
Gamme d'assemblage	Ligne 1 – Type A	Ligne 2 – Type B
Opération 1	Placement de la colle sur la carte vierge	Placement de la colle sur la carte vierge
Opération 2	Placement des CMS par un robot	Placement des CMS par un robot
Opération 3	Placement manuel des composants Comp1, Comp2 et Comp3	Placement manuel des composants Comp11, Comp12, Comp13, Comp14 et Comp2
Opération 4	Placement manuel des composants Comp4, Comp5 et Comp6	Placement manuel des composants Comp1, Comp4, Comp10 et Comp8
Opération 5	Passage sur vague d'étain	Passage sur vague d'étain

Opération 6	Contrôle visuel par comparaison	Contrôle visuel par comparaison
Opération 7	Contrôle automatique de ses fonctionnalités	Contrôle automatique de ses fonctionnalités

Tableau 4.1. Gammes d'assemblage des cartes électroniques de type A et B.

Le plan schématique de l'atelier peut être représenté comme ci-dessous (Figure 4.8) sachant que les convoyeurs entre les postes ne sont pas représentés et la zone verte correspond aux zones piétonnes.

Ligne 1 - Type A



Ligne 2 - Type B

Figure 4.8. Configuration de l'atelier de deux lignes d'assemblage (A et B).

Pratiquement, plusieurs problématiques peuvent être rencontrées par ces configurations. Nous en citons quelques cas illustrant la nécessité et le besoin de cette entreprise et ses lignes d'assemblage d'être reconfigurables et agiles :

- L'entreprise souhaite implanter une nouvelle ligne d'assemblage polyvalente, mais n'a pas la place suffisante pour installer une troisième ligne ;
- La demande de cartes de type A est très forte et la capacité de production de la ligne 1 n'est pas suffisante. L'entreprise souhaite rendre les 2 lignes polyvalentes afin que les assemblages des cartes électroniques de type A et B soient possibles sur les deux lignes ;
- On souhaite former une seule ligne d'assemblage au centre à partir des postes des deux premières lignes.

Par exemple, supposons que l'entreprise souhaite produire un nouveau type C de cartes électroniques, mais n'a pas la place suffisante pour installer une troisième ligne. Voici sa gamme d'assemblage (Tableau 4.2) :

Gamme d'assemblage	Ligne 3 – Type C
Opération 1	Placement de la colle sur la carte vierge
Opération 2	Placement des CMS par un robot
Opération 3	Placement manuel des composants Comp7, Comp8 et Comp2
Opération 4	Placement manuel des composants Comp9, Comp10 et Comp4
Opération 5	Placement manuel des composants Comp5, Comp8 et Comp9
Opération 6	Passage sur vague d'étain
Opération 7	Contrôle visuel par comparaison
Opération 8	Contrôle automatique de ses fonctionnalités

Tableau 4.2. Gamme d'assemblage des cartes électroniques de type C.

Afin de résoudre ces problèmes, nous proposons d'adapter le concept de SPRA aux deux lignes :

☞ Postes de travail/robots modulaires : Les étapes d'assemblage se réalisent sur des modules de format identique. D'un module à l'autre, seule la fonction change.

☞ Les modules sont mobiles ou déplaçables à l'aide d'un robot AGV, par exemple en forme de quadrillage.

☞ Les modules sont reliés entre eux par des convoyeurs.

La Figure 4.9 suivante représente la nouvelle configuration. Les zones piétonnes sont désormais placées de part et d'autre des lignes. Au centre, c'est une zone

destinée à accueillir des nouveaux modules déplaçables. On note que l'on est ici dans une nouvelle configuration d'assemblage des cartes de type A et B.

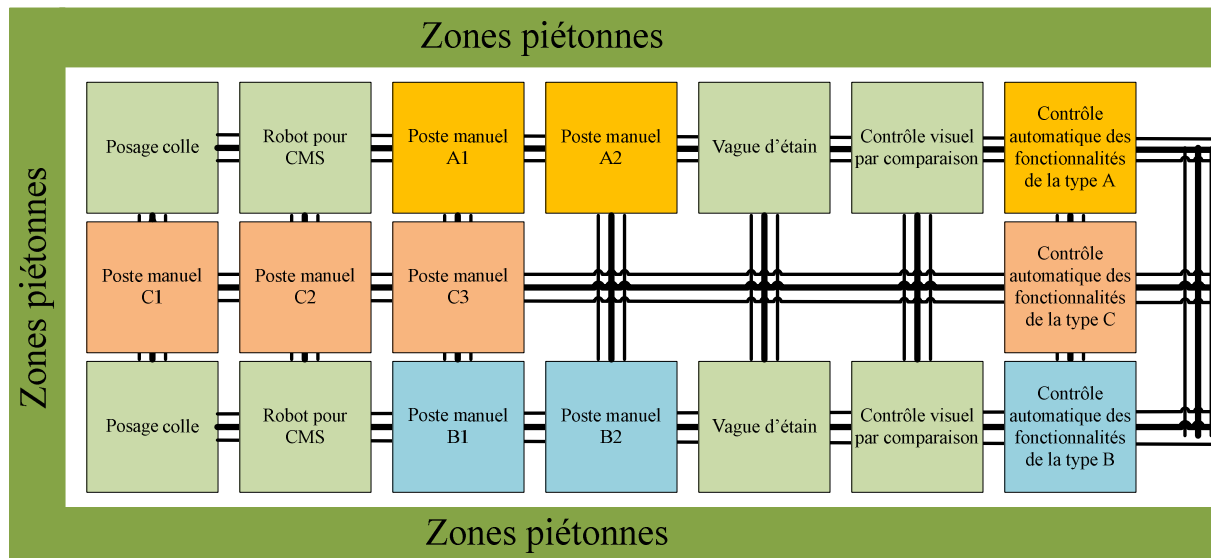


Figure 4.9. Nouvelle configuration de l'atelier de deux lignes d'assemblage (A et B).

Pour montrer l'applicabilité de notre proposition en adaptant le concept de SPRA à cette entreprise et bien sûr à ses lignes d'assemblage, nous considérons que l'état du marché exige le passage de cartes A et B aux cartes A et C.

Tout d'abord, la ligne 2 (type B) s'arrête. Nous repositionnons ensuite les modules de l'atelier à l'aide des rails comme illustré au dessous (Figure 4.10). Ainsi, la ligne 2 est prête pour l'assemblage de cartes de type C.

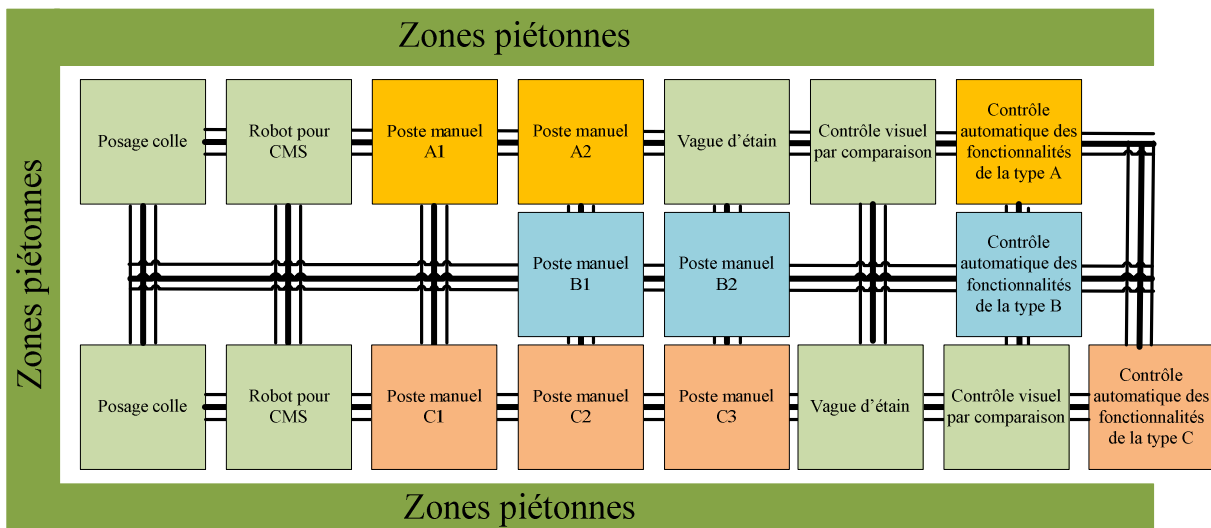


Figure 4.10. Nouvelle configuration de l'atelier de deux lignes d'assemblage (A et C).

Cet exemple est illustré de façon assez simplifiée. Nous allons désormais pousser plus loin le déploiement d'une nouvelle configuration lorsque nous aurons besoin de passer d'une configuration X à une autre configuration Y répondant à la demande des clients et aux exigences du marché.

5. Projection de nos approches

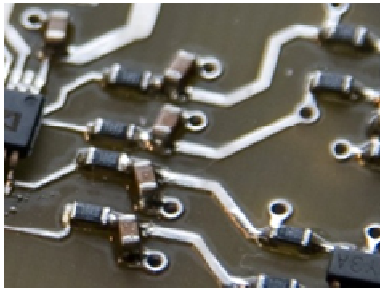
Dans cette section, nous projetons nos approches proposées dans ce mémoire, notamment le modèle générique et la méthodologie de reconfiguration, sur une ligne l'assemblage de cartes électroniques pour montrer l'intérêt de cette étude, évaluer nos solutions et mettre notre étude dans un contexte réel.

Tout d'abord, nous projetons le modèle générique proposé dans le premier chapitre de la deuxième partie en distinguant ses quatre parties (structure physique, structure logique, configuration physique, configuration logique) et les opérations qui jouent un rôle central puisqu'elles les relient.

Parallèlement, nous suivons la méthodologie proposée dans la troisième partie. Cette méthodologie définit les étapes à suivre pour concevoir et implanter une nouvelle configuration. Effectivement, nous intégrons le processus de reconfiguration avec le modèle générique.

Prenons pour exemple le cas d'une entreprise qui réalise sur une ligne de production l'assemblage de trois types des cartes électroniques. Les compétences de l'entreprise (**Aube, 2013**) lui permettent de réaliser de nombreuses opérations sur les cartes imprimées (Figure 4.11).

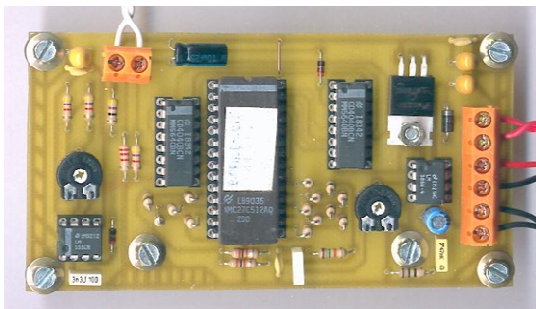
Soudage de composant CMS
(Composant Monté en Surface)



Soudage de composant simple
(manuellement et automatiquement)



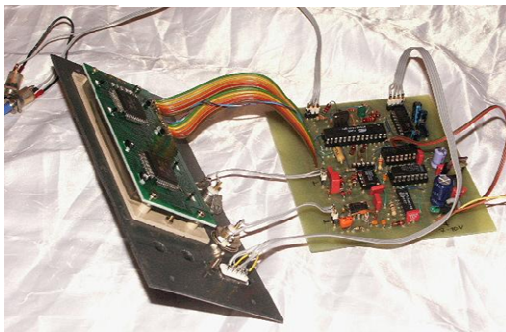
Montage de composants



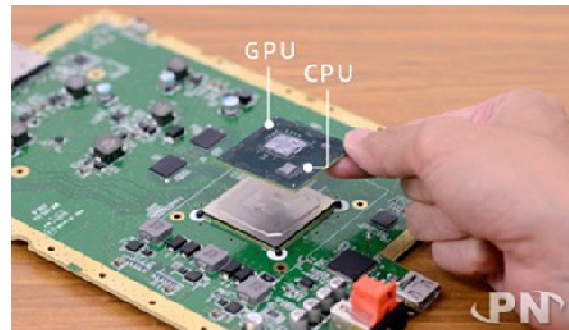
Câblage de fils



Assemblage de cartes



Dépose de colle



Dépose de film protecteur + Gravure



Contrôle



Figure 4.11. Compétences d'une entreprise.

La Figure 4.12 illustre les différents éléments disponibles, en général. Les composants de la **structure physique** (S_p) de cette ligne (Figure 4.13) sont les ressources stationnaires (R_s), les ressources de transport (R_t) ainsi que les différentes connexions ($Conn$) dont chacune relie deux ressources stationnaires entre elles par l'intermédiaire d'une ressource de transport qui réalise le transfert, ces connexions sont représentées par des flèches.

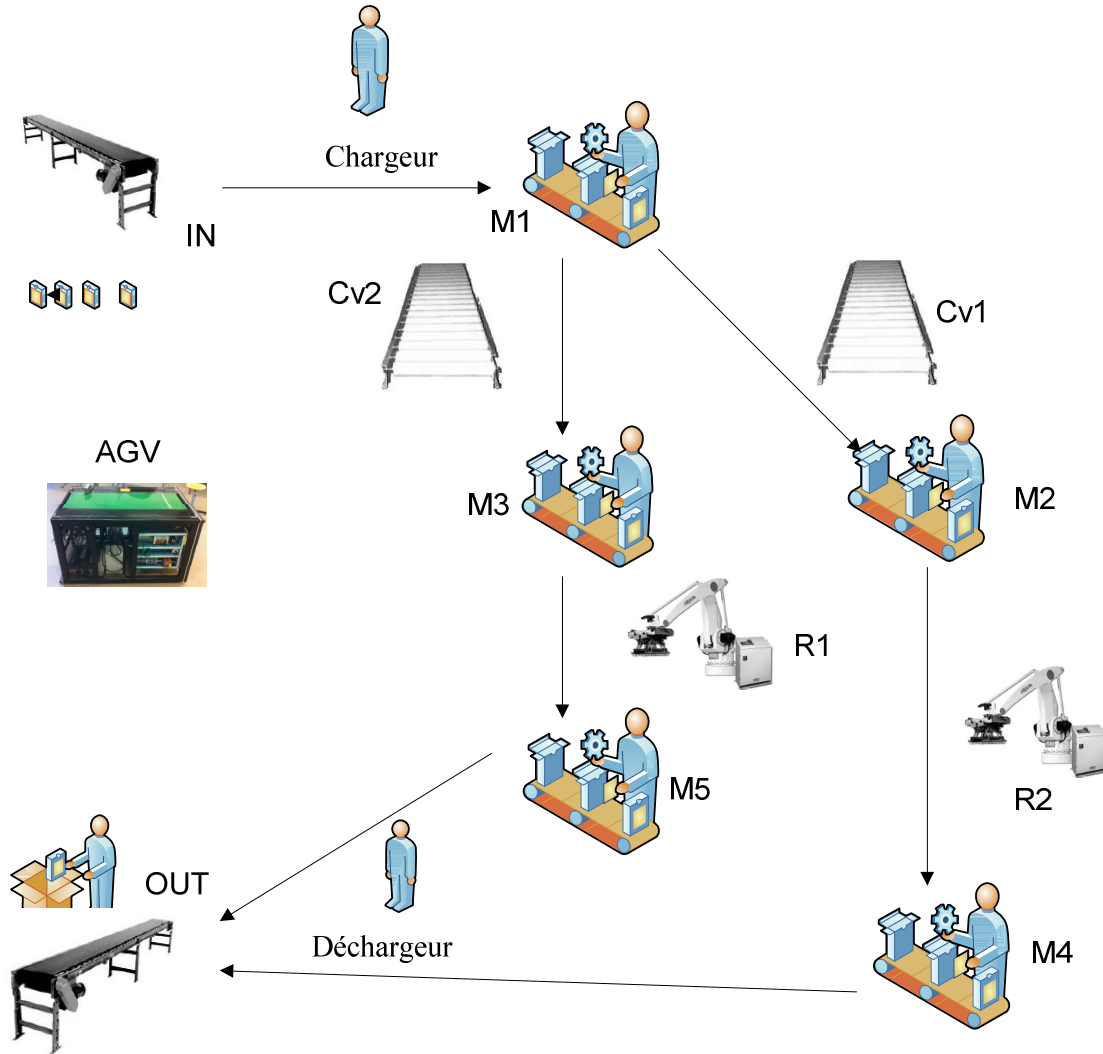


Figure 4.12. Structure physique de la ligne d'assemblage : cas général.

$$R_s = \{IN, OUT, M1, M2, M3, M4, M5\}$$

$$R_t = \{chargeur, déchargeur, R1, R2, Cv1, Cv2\}$$

$$R = \{R_s, R_t\}$$

$$S_p = \{R, Capab, Conn\}$$

$Capab$ décrit les capacités de l'ensemble des ressources (R) qui peuvent être définies comme suit :

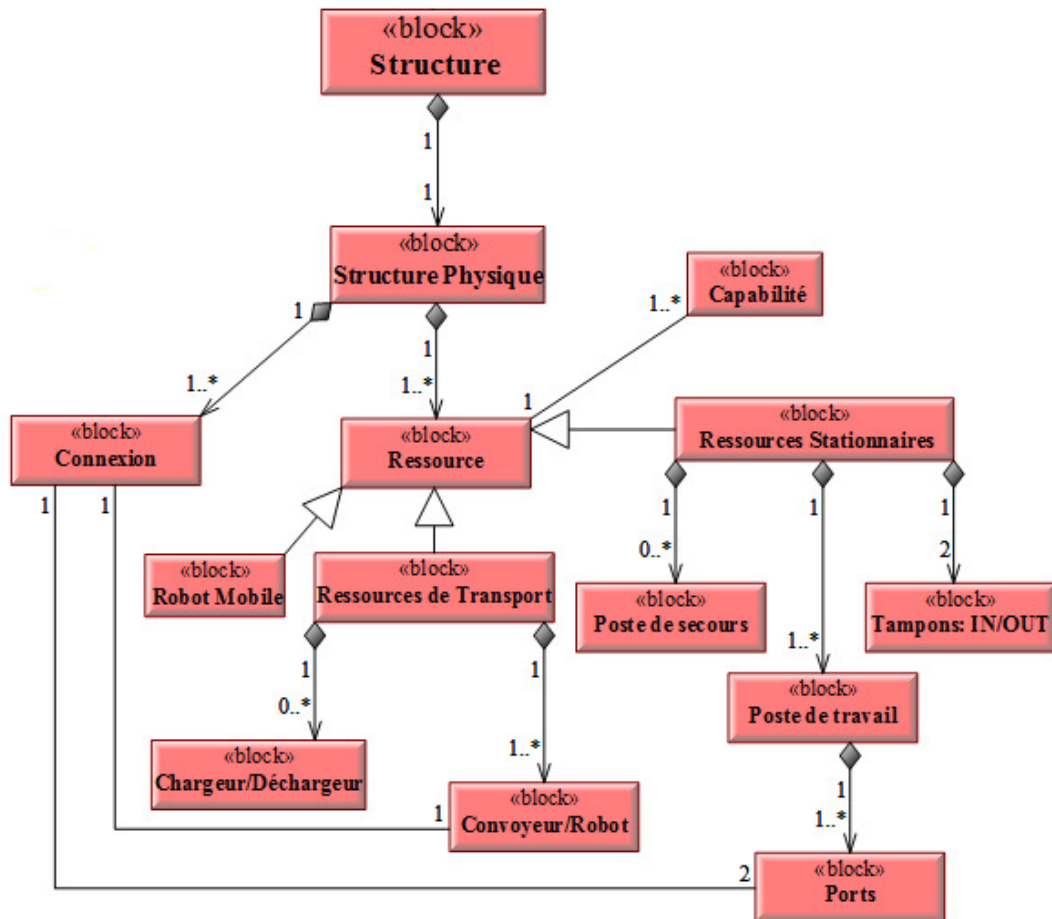


Figure 4.13. Composants de la structure physique-cas d'étude.

- IN, OUT: ces deux tampons représentent l'entrée et la sortie du système et sont modélisés comme des ressources de stockage;
- M1 : ce poste de travail est capable de réaliser les opérations suivantes :
 Soudage de composants CMS ;
 Soudage manuel de composants électroniques simples ;
 Soudage automatique de composants électroniques simples ;
 Montage de composants ;
 Contrôle soudage.
- M2 : cette ressource peut réaliser :
 Soudage manuel de composants électroniques simples ;
 Dépose de film protecteur ;
 Câblage de fils ;
 Contrôle soudage.
- M3 : il s'agit d'une machine capable de faire :

Montage de composants ;

Câblage de fils ;

Contrôle soudage ;

Gravure.

- M4 : cette ressource peut réaliser :

Soudage de composants CMS ;

Soudage automatique de composants électroniques simples ;

Câblage de fils ;

Contrôle final.

- M5 : cette machine est capable de faire :

Câblage de fils ;

Dépose de film protecteur ;

Dépose d'étiquette ;

Gravure ;

Contrôle final.

- Chargeur : capable d'assurer les transports entre IN et les ressources de la première phase de la ligne d'assemblage ;

- Déchargeur : capable d'assurer les transports entre les ressources de la dernière phase de cette ligne d'assemblage et OUT. Le chargeur et le déchargeur peuvent être assurés par le robot AGV ;

- R1, R2 : robots capables de réaliser toutes les connexions. Dans certains cas, l'AGV peut remplacer un ou les deux robots si c'est nécessaire.

- Cv1, Cv2 : convoyeurs capables d'effectuer uniquement les transferts entre deux postes voisins et dans les deux directions. Le robot AGV peut aussi réaliser ces tâches en cas de besoin. Dans notre cas, les postes sont équipés de convoyeurs.

La structure logique (S_l) est constituée de la façon suivante (Figure 4.14) :

- Produits : $Pr = \{Pr1, Pr2, Pr3\}$ trois types de cartes électroniques.
- Gammes logiques : $G = \{G1, G2, \dots, G5\}$ cinq gammes logiques permettent d'obtenir ces trois produits en réalisant certaines fonctions de travail.

- Fonctions : $F = \{F_{tr}, F_{ts}, F_{st}\}$

$F_{tr} = \{F_{tr1}, F_{tr2} \dots F_{tr20}\}$ représentent les fonctions de travail sachant que chacune doit être implémentée au moins une fois par opération et que chaque poste peut réaliser plusieurs opérations. Par exemple :

F_{tr1} : soudage de composants CMS.

F_tr2 : soudage manuel de composants électroniques simples.

F_tr3 : soudage automatique de composants électroniques simples.

F_tr4 : contrôle soudage.

F_tr5 : montage de composants.

F_tr6 : câblage de fils.

...

F_tr17 : dépose de film protecteur.

F_tr18 : gravure.

F_tr19 : dépose d'étiquette.

F_tr20 : contrôle final.

$F_{ts} = \{F_{ts1}, \dots, F_{ts4}\}$ sont les fonctions de transport dont chacune est implémentée par une seule opération qui est reliée à une connexion. Par exemple :

F_ts1 : de IN à M1.

F_ts2 : de M1 à M3.

F_ts3 : de M3 à M5.

F_ts4: de M5 à OUT.

$F_{st} = \{F_{st_in}, F_{st_out}\}$ représentent les fonctions de stockage dont chacune est implémentée par une seule opération :

F_st_in : permet de stocker le produit brut Pr0 sur le tampon IN.

F_st_out : permet de stocker le produit final Prj sur le tampon OUT.

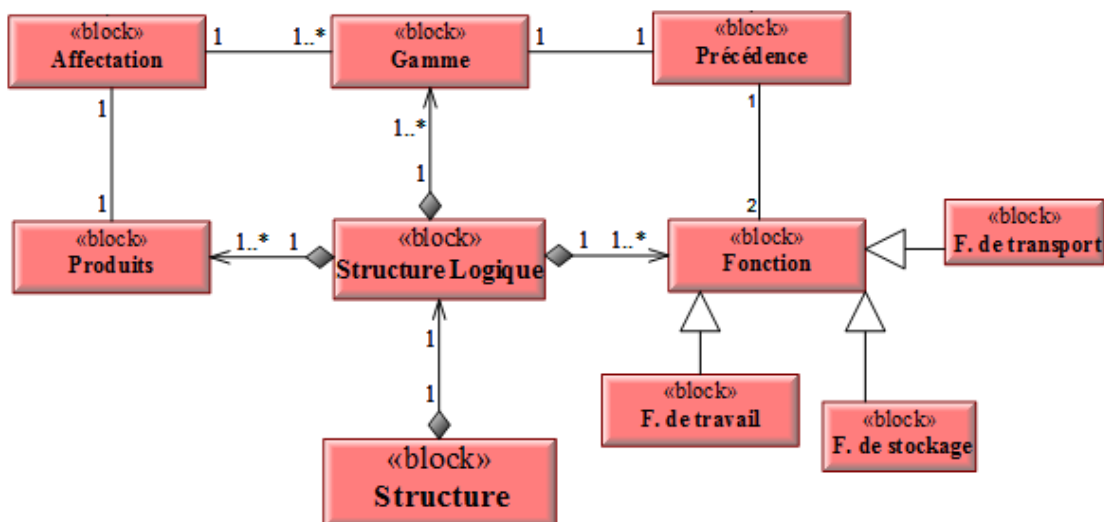


Figure 4.14. Composants de la structure logique-cas d'étude.

Ensuite, nous affectons les gammes logiques aux produits en supposant que certains produits peuvent être réalisés à partir de plusieurs gammes logiques. Ainsi, nous définissons l'application *Affec*. Par exemple :

- G1, G2 : Pr1.
- G3, G4 : Pr2.
- G5 : Pr3.

Enfin, nous définissons les relations de précédences (*Prec*) entre les fonctions de travail au sein des gammes logiques :

- G1 = {F_tr1, F_tr2, F_tr4, F_tr5, F_tr6, F_tr17 F_tr18, F_tr19, F_tr20}.
- G2 = {F_tr1, F_tr3, F_tr4, F_tr5, F_tr6, F_tr17 F_tr18, F_tr19, F_tr20}.
- G3 = {F_tr2, F_tr4, F_tr10, F_tr17, F_tr18, F_tr12, F_tr20}.
- G4 = {F_tr3, F_tr10, F_tr17, F_tr18, F_tr12, F_tr20}.
- G5 = {F_tr1, F_tr2, F_tr10, F_tr6, F_tr18, F_tr20}.

La structure logique (S_l) est donc définie par :

$$S_l = \{F, G, Pr, Prec, Affec\}$$

Par conséquent, **la structure (S)** du SPRA est donnée par la paire :

$$S = \{S_p, S_l\}$$

Nous rappelons que nous pouvons faire correspondre à une même structure un ensemble de configurations différentes dont chacune définit les différentes utilisations et organisations des éléments de la structure pour que cette configuration atteigne son objectif. Les configurations du système évoluent ainsi selon ses objectifs et chaque configuration doit répondre au moins à un objectif ou un besoin.

En pratique, la fonction de la reconfiguration démarre le plus souvent du **besoin de lancer un nouveau produit**. Par exemple pour **Pr1**, nous choisissons la gamme logique correspondante, soit **G1**. Il est à rappeler et à noter que plusieurs opérations de la gamme peuvent être réalisées sur un même poste. Par exemple, une opération de soudage de composants CMS et une opération de contrôle sont affectées au même poste qui doit en être capable.

Nous choisissons ensuite **les ressources capables** de réaliser l'ensemble des fonctions de cette gamme, soient les ressources utilisées : $R_{util} = \{IN, OUT, M1, M3, M5, chargeur, AGV, R1\}$. Effectivement, cela représente **la première partie de la configuration physique (C_p)**, (Figure 4.16).

Pour réaliser la gamme, nous définissons **une organisation de ces ressources** et aussi les différentes connexions entre elles. Par exemple :

N = 3 postes de travail.

m = 3 phases.

Nous présentons le poste de travail de la phase i et du numéro local pour la phase j par $M_x = P[i, j]$. Nous avons alors : $M1 = P[1, 1]$, $M3 = P[2, 1]$, $M5 = P[3, 1]$ et nous leur associons leurs coordonnées cartésiennes.

Ensuite, nous définissons **les connexions** (l'ordonnancement des postes) comme suit :

$IN.PostesSuivants = P[1, 1]$, cela représente la connexion $IN_P[1, 1]$.

$P[1, 1].PostesSuivants = P[2, 1]$, il s'agit de la connexion $P[1, 1]_P[2, 1]$.

$P[2, 1].PostesSuivants = P[3, 1]$, c'est la connexion $P[2, 1]_P[3, 1]$.

$P[3, 1].PostesSuivants = OUT$, cela représente la connexion $P[3, 1]_OUT$.

Maintenant, nous définissons **les opérations de l'assemblage** qui sont représentées par six types d'opérations (Figure 4.15) :

- Opérations principales : il s'agit d'une instance d'une fonction de travail à un poste associé :

$Op1(M1, F_{tr1})$: soudage de composants CMS.

$Op2(M1, F_{tr2})$: soudage manuel de composants électroniques simples.

$Op3(M3, F_{tr4})$: contrôle soudage.

$Op4(M3, F_{tr5})$: montage de composants.

$Op5(M3, F_{tr6})$: câblage de fils.

$Op6(M5, F_{tr17})$: dépose de film protecteur.

$Op7(M5, F_{tr18})$: gravure.

$Op8(M5, F_{tr19})$: dépose d'étiquette.

$Op9(M5, F_{tr20})$: contrôle final.

- Opérations de transport : chaque opération est reliée à une connexion et à une instance de fonction de transport correspondante :

$Op_{ts1} : IN_P[1, 1]$, cette opération est réalisée par le chargeur.

$Op_{ts2} : P[1, 1]_P[2, 1]$, elle est réalisée par les convoyeurs des postes $M1$ et $M3$.

$Op_{ts3} : P[2, 1]_P[3, 1]$, elle est réalisée par le robot $R1$.

$Op_{ts4} : P[3, 1]_OUT$, cette opération est réalisée par le robot mobile AGV.

- Opérations de stockage : elles assurent de stocker les produits sur les deux tampons IN et OUT :

Op_{st_in} : le produit brut $Pr0$ (circuit imprimé) est disponible sur IN .

Op_{st_out} : le produit final $Pr1$ (carte électronique assemblée type 1) est stockée sur OUT .

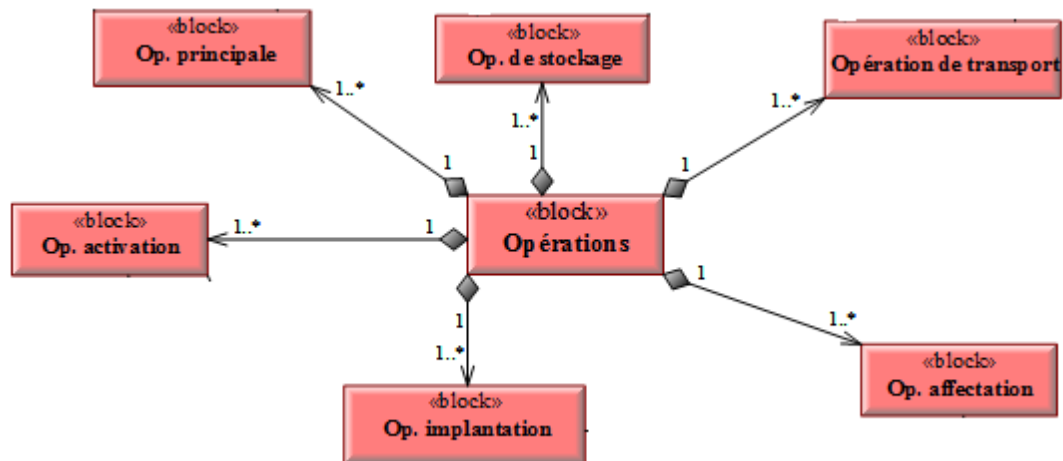


Figure 4.15. Composants des opérations-cas d'étude.

- Opérations d'affectation : elles permettent d'affecter les ressources utilisées dans cette configuration. Donc, nous affectons seulement les ressources utilisées qui sont M1, M3, M5, R1 et AGV et nous les mettons en place.
- Opérations d'activation : elles représentent **la deuxième partie de la configuration physique (C_p)** (Figure 4.16) parce que ces opérations permettent d'activer des modules (outils et/ou dispositifs) sur les ressources utilisées selon les exigences de la configuration demandée ($Capab_assoc$). Nous activons tout ce qui sera utilisé et en même temps nous désactivons les autres. Dans notre cas :

Sur le poste M1 : nous activons les dispositifs capables de faire le soudage des composants CMS et le soudage manuel de composants électroniques simples.

Sur le poste M3 : nous activons les dispositifs capables de faire le contrôle du soudage, le montage de composants et le câblage de fils.

Sur le poste M5 : nous activons les dispositifs capables de déposer le film protecteur, de faire la gravure, de déposer l'étiquette et de faire le contrôle final.

Pour le robot R1 : nous activons ce robot pour réaliser uniquement la connexion de M3 à M5.

Pour le robot AGV : nous demandons de ce robot de transporter uniquement les produits finaux de M5 à OUT.

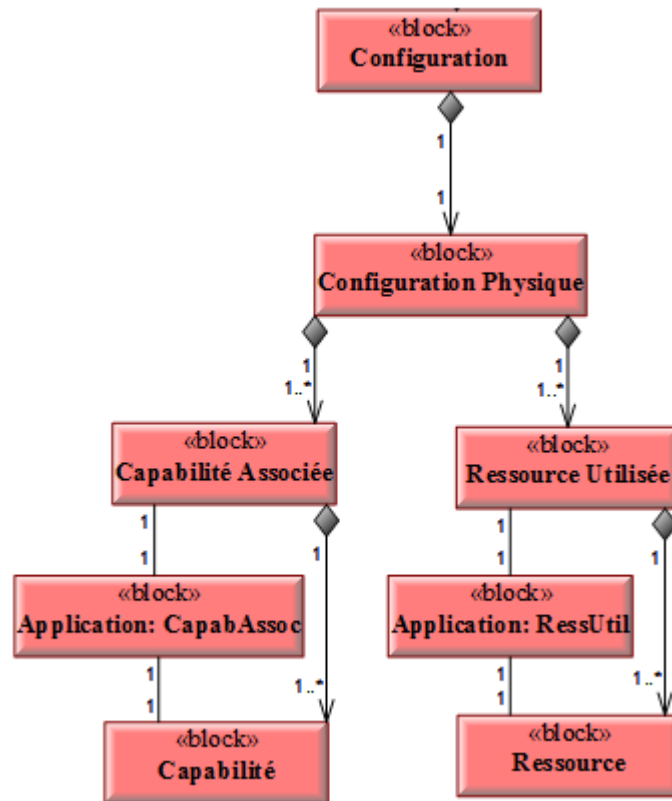


Figure 4.16. Composants de la configuration physique-cas d'étude.

- Opérations d'implantation : elles représentent **la configuration logique (C_l)**. Elles permettent d'implanter des programmes de contrôle sur chaque ressource utilisée dans la configuration actuelle (Figure 4.17). Autrement dit, nous envoyons les programmes de contrôle compatibles avec la configuration actuelle à chaque ressource utilisée.

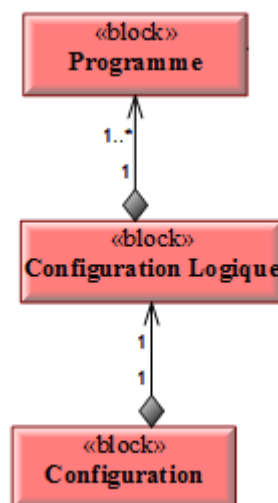


Figure 4.17. Composants de la configuration logique-cas d'étude.

Enfin, nous mettons une séquence d'opérations compatible avec cette gamme logique : Op_st_in, Op_ts1, Op1, Op2, Op_ts2, Op3, Op4, Op5, Op_ts3, Op6, Op7, Op8, Op9, Op_ts4, Op_st_out.

Par conséquent, cette première configuration peut être illustrée sur la figure suivante (Figure 4.18).

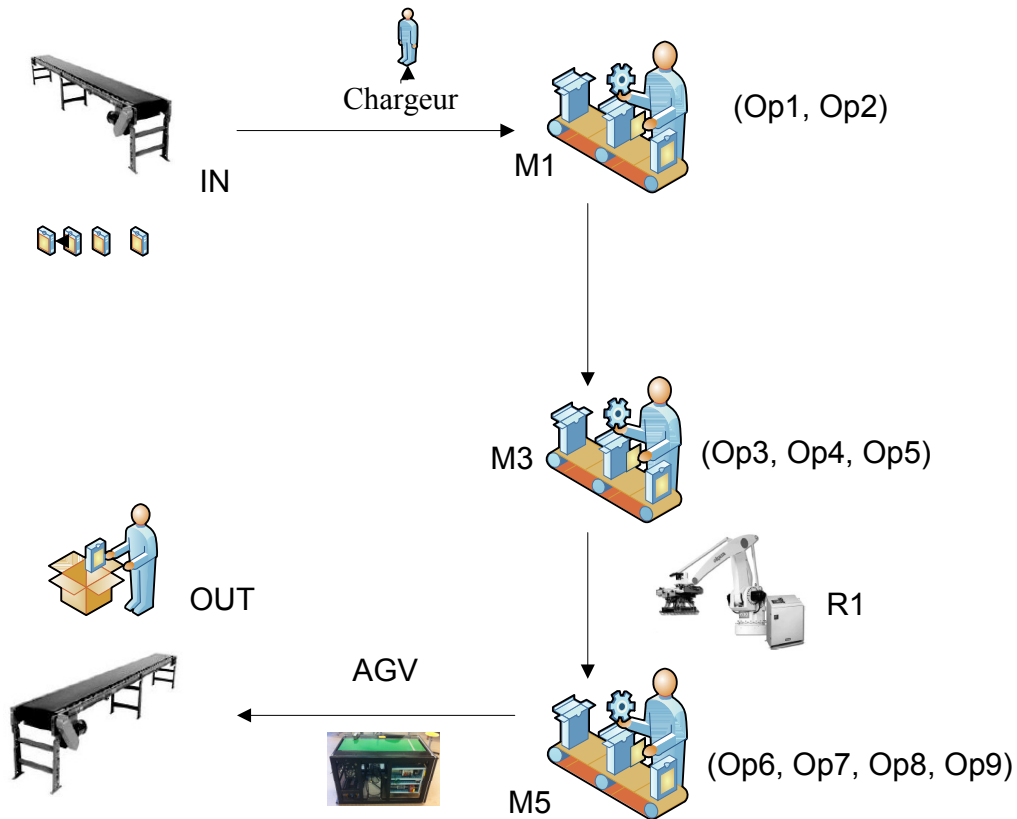


Figure 4.18. Configuration 1 de cette ligne d'assemblage.

En résumé, la méthode commence tout d'abord par la définition de la structure logique selon le besoin (le produit et sa gamme logique) qui reconfigure le système et définit ainsi la structure physique correspondante (les ressources capables et les différentes connexions entre elles). Parallèlement et selon cette structure physique, la structure logique reconfigure le système et définit la configuration logique (les programmes de contrôle) qui, à son tour, reconfigure le système et définit finalement la configuration physique (les ressources utilisées et les capacités demandées de chaque ressource) (Figure 4.19).

Effectivement, il n'y a pas de frontières bien définies entre ces quatre composants (structure physique, structure logique, configuration physique, configuration logique). Autrement dit, nous remarquons qu'il y a d'intersections claires entre ces composants. Par exemple, les ressources utilisées R_{util} dans une configuration sont incluses dans la structure physique et en même temps elles définissent la première partie de la configuration physique.

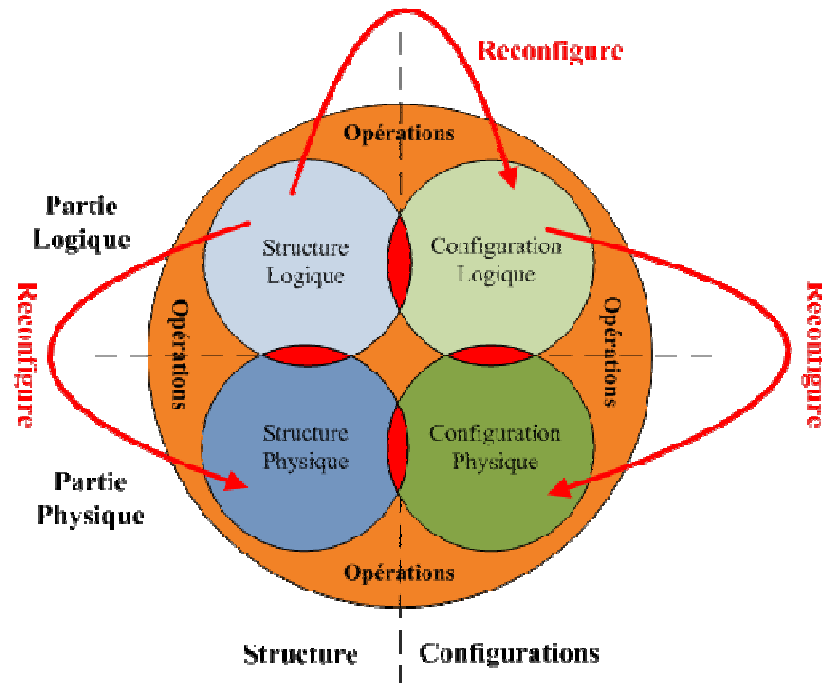


Figure 4.19. Méthodologie de la reconfiguration_cas d'étude.

Par contre, **si une panne survenait**, par exemple sur la machine M3, habituellement un arrêt de la ligne d'assemblage aurait été réalisé pour réparer la machine, puis pour ensuite continuer le plan de production. L'échéancier se retrouvait donc décalé du temps de réparation. Ainsi, la reconfiguration de la ligne d'assemblage assure une continuité du service avec une nouvelle organisation de la ligne afin de répondre aux besoins du client dans les délais impartis.

Le modèle générique et la méthodologie proposée dans ce mémoire permettent de reconfigurer rapidement et efficacement cette ligne en modifiant les ressources actuellement utilisées, leurs coordonnées, leurs capacités et leurs programmes de contrôle associés pour chaque composant de la nouvelle structure. Par exemple et tout simplement, nous désaffectons le poste M3 qui est en panne, dans ce cas $R_{util} = \{IN, OUT, M1, M5, chargeur, AGV, R1\}$, nous modifions la mission du robot R1 et nous activons de nouveau les dispositifs sur les deux postes M1 et M5 comme suit :

Sur le poste M1 : nous activons les dispositifs capables de faire le soudage des composants CMS et le soudage manuel de composants électroniques simples. En plus, nous installons de nouveau les dispositifs capables de faire le contrôle du soudage et le montage de composants (déjà réalisés sur la machine M3), et nous activons ensuite ces dispositifs.

Sur le poste M5 : nous activons les dispositifs capables de déposer le film protecteur, de faire la gravure, de déposer l'étiquette et de faire le contrôle final. En

plus, nous installons et activons le dispositif capable de faire le câblage de fils (déjà réalisé sur la machine M3).

Pour le robot R1 : nous activons ce robot pour réaliser uniquement la connexion de M1 à M5.

Il est à rappeler que les opérations déjà demandées du poste M3 sont demandées de nouveau des postes M1 et M5 qui sont capables de les faire.

Finalement, nous implantons de nouveau les programmes de contrôle et nous définissons également une nouvelle séquence d'opérations. La figure suivante illustre cette nouvelle configuration (Figure 4.20).

Ainsi, conduire un système reconfigurable et agile nous a permis d'être réactifs avec cette situation brusque (M3 en panne) sur la ligne d'assemblage, et d'améliorer la qualité du service rendu en respectant les délais de livraison pour satisfaire le client.

Dans la suite de cette dernière partie, nous présentons quelques travaux pratiques qui ont servi à reconfigurer le système et à déployer une configuration.

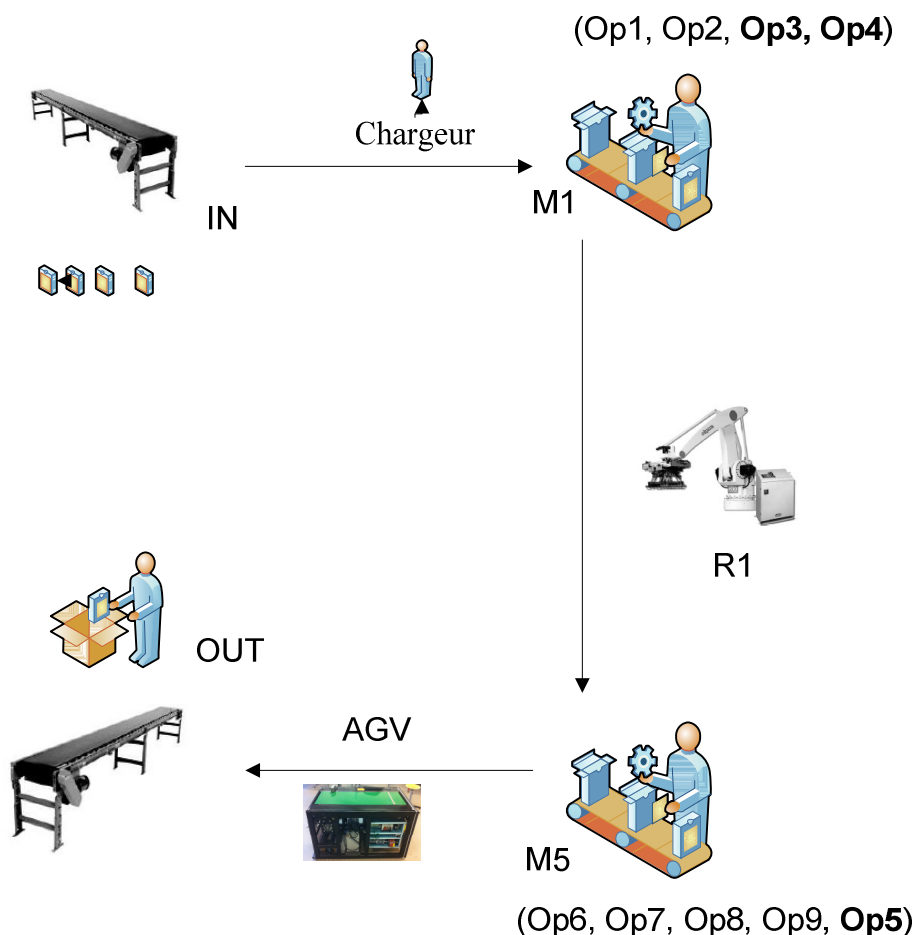


Figure 4.20. Configuration 2 de cette ligne d'assemblage.

6. Interfaces de reconfiguration

Tout d'abord, nous avons besoin de créer une interface principale qui permet au configurateur/chef de l'atelier de réaliser les cinq fonctions suivantes :

☒ Choisir une configuration : cette fonction permet de choisir une configuration parmi celles déjà enregistrées.

☒ Saisir une nouvelle configuration : elle permet à l'utilisateur d'enregistrer une nouvelle configuration. Pour cela, l'utilisateur est guidé dans ces choix, et les champs ne s'affichent que s'ils ont besoin d'être remplis suivant les paramètres de chaque configuration.

☒ Visualiser la configuration : cela donne une représentation graphique de la configuration. Ainsi, il est possible de faire afficher des formes représentant les postes de travail, puis de les relier par des flèches.

☒ Évaluer la configuration : cette fonction appuie principalement sur le calcul d'indicateurs de performance (le temps opératoire de chaque configuration, le rapport entre le temps opératoire et du temps total, le coût, la fiabilité de la configuration, ...). Ainsi, un système d'aide à la décision permet au responsable de la ligne de choisir la meilleure configuration.

☒ Implanter la configuration : elle permet de récupérer depuis une base de données tout ce qui est nécessaire pour une configuration donnée (les capacités, les ordonnancements des postes utilisés, ...).

En fait, cette interface principale a été réalisée comme illustrée sur la figure suivante (Figure 4.21). Elle donne accès à plusieurs sous interfaces correspondant chacune à une des actions à réaliser par notre système. Certaines sous interfaces ne sont pas disponibles tant qu'une configuration n'a pas été encore choisie ou conçue.

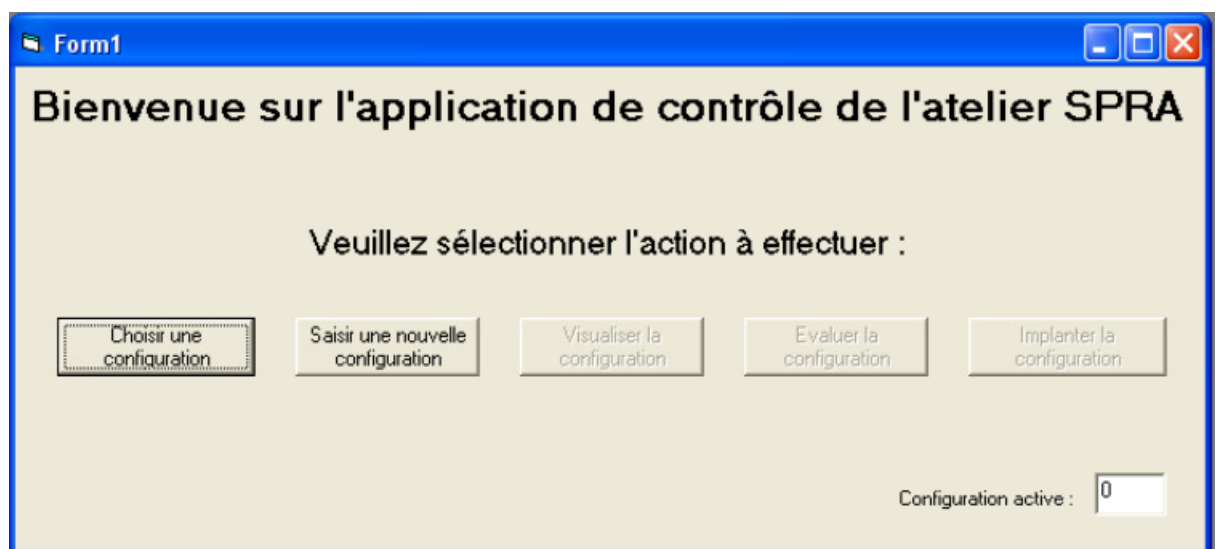


Figure 4.21. Interface principale de reconfiguration.

Premièrement à gauche, nous trouvons la sous interface «Choisir une configuration» qui nous permet de choisir une configuration déjà enregistrée dans la base de données. Ensuite, nous trouvons la sous interface «Saisir une nouvelle configuration» qui nous permet d'ajouter une nouvelle configuration et enrichir ainsi la base de données. Effectivement, ces deux sous interfaces traduisent le bloc « Choix/Construction une configuration » introduit dans le deuxième chapitre de la deuxième partie (Figure 4.22).

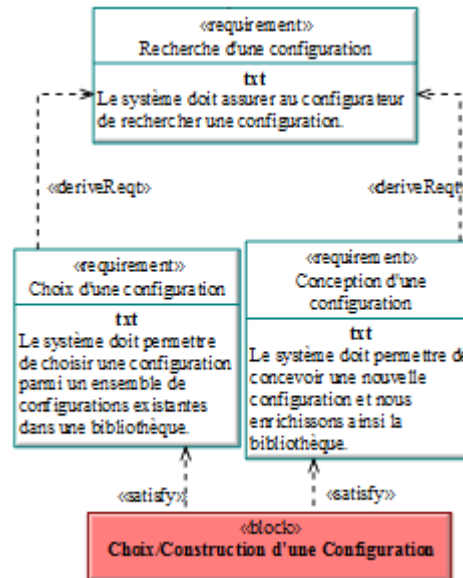


Figure 4.22. Satisfaction des exigences par un bloc (interfaces)_cas d'étude.

Pour cela, à partir des différentes formalisations de la structure et de la configuration déjà proposées dans ce mémoire, nous avons pu construire une interface de saisie d'une configuration qui sera donc représentée dans une base de données qui contient toutes les informations nécessaires. Ainsi, nous pouvons utiliser ce fichier pour visualiser cette configuration (affichage de l'organisation de l'atelier), pour l'évaluer et également pour l'implanter.

La Figure 4.23 illustre tous les paramètres qui permettent de définir une configuration comme nous l'avons détaillé dans le premier chapitre de la deuxième partie. Nous rappelons que chaque configuration peut être définie par les paramètres suivants : le nombre de postes de travail N , le nombre de phases m , le nombre de postes à chaque phase $np[m]$, l'arrangement de postes, les connexions entre eux, les coordonnées cartésiennes (x_i, y_i) pour chaque poste au niveau de l'atelier, et les capacités demandées de chaque poste utilisé dans cette configuration.

Form2

Numéro de configuration à ajouter: 12 Nombre de postes: 9 Nombre de phases: 3

Ajouter cette configuration

Numéro de poste	CAPABILITES									Xi	Yi	N° phase	Postes suivants
	cap1	cap2	cap3	cap4	cap5	cap6	cap7	cap8	cap9				
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Valider cette configuration

Figure 4.23. Sous interface «Saisir une nouvelle configuration».

Les deux sous interfaces « Visualiser la configuration » et « Evaluer la configuration » ne font pas partie de notre application.

Par ailleurs, la sous interface « Implanter la configuration » permet de récupérer depuis la base de données toutes les valeurs des capacités et des ordonnancements des postes utilisés dans une configuration donnée, et d'envoyer ensuite les programmes de contrôle correspondants aux ressources utilisées.

Au niveau de l'organisation de la base de données, nous avons utilisé une base SQL Server pour gérer les configurations de l'atelier. Cette base est organisée en deux parties distinctes :

- ✓ La première contient toutes les configurations avec le nombre de postes assignés (N), le nombre de phase (m), le nombre de poste par phase (np) ainsi que sa référence.
- ✓ La seconde contient les binômes configuration-poste, avec les capacités assignées, l'ordre des opérations, la position du poste dans l'atelier pour cette configuration ainsi que les postes suivants possibles pour assembler/fabriquer ce produit.

En ce qui concerne l'implantation d'une configuration, nous avons créé les interfaces nécessaires Pro-face, par exemple une interface du poste 1 pour la gamme logique 2 peut afficher l'ordre des opérations avec les images correspondantes. En plus, une autre interface proposée peut afficher d'autres informations nécessaires, par exemple chaque opérateur peut faire l'identification sur le poste (son nom et son code référentiel), il peut signaler dès qu'il est prêt pour commencer le travail, l'opérateur peut également échanger des messages avec le superviseur pour informer de défauts, de problèmes techniques ou en cas de besoin d'arrêter le travail sur ce poste, et il peut aussi compter le nombre de pièces traitées en distinguant les bonnes et les mauvaises pièces (Figure 4.24). Effectivement, nous n'affichons sur l'écran de l'opérateur que les informations désirées en respectant l'ordre des actions demandées.

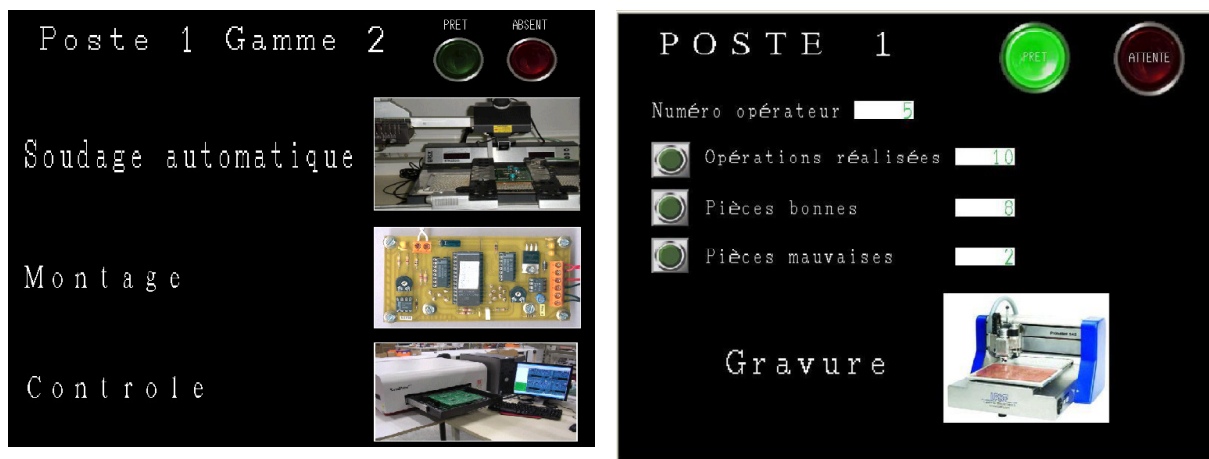


Figure 4.24. Interfaces Pro-face pour l'opérateur.

Enfin, au niveau de la supervision, l'interface de la supervision doit permettre d'afficher les informations globales de la configuration, par exemple : la référence du produit en cours, la gamme logique, nombre de postes, nombre d'opérations, et également cette interface doit afficher les informations détaillées pour chaque ressource, par exemple : l'état de chaque ressource (fonctionnement, arrêt), nom de l'opérateur ou sa référence, état de l'opérateur (prêt ou non), opération en cours, nombre de pièces bonnes, nombre de pièces rejetées, durée de fonctionnement, durée d'arrêt, temps moyen par pièce. En plus, cette interface donne au superviseur une visualisation globale de l'atelier affichant l'organisation des ressources utilisées dans une configuration donnée, et permet aussi d'échanger des messages avec les opérateurs (Figure 4.25).

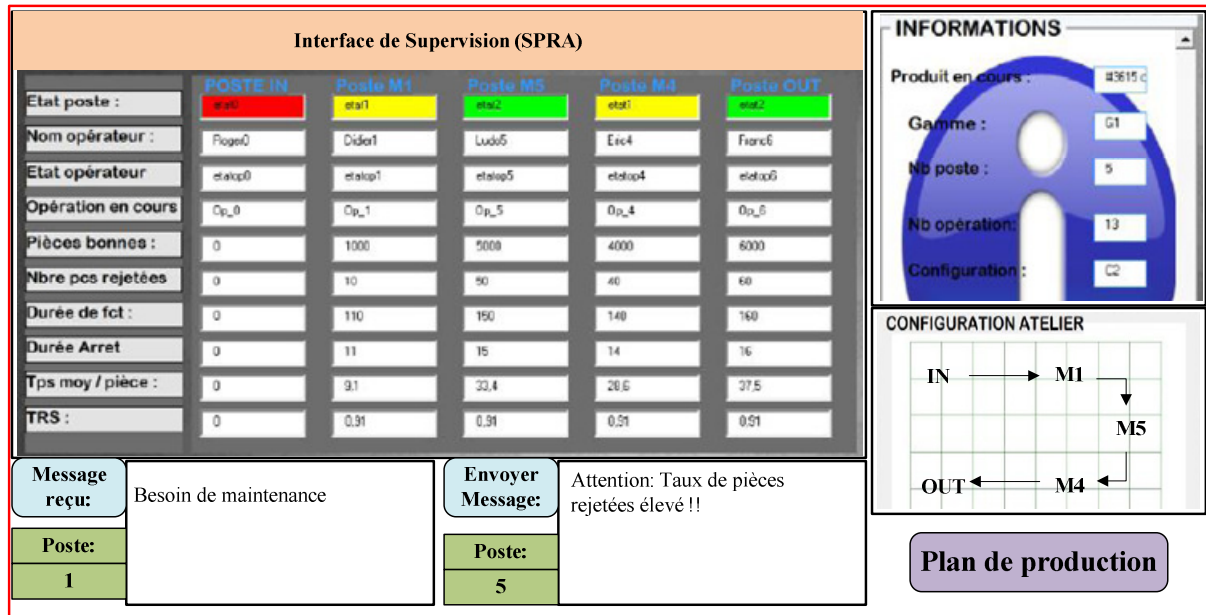


Figure 4.25. Interface de supervision.

7. Conclusion

Dans cette dernière partie du mémoire, nous avons essayé de montrer l'applicabilité de nos approches sur la ligne d'assemblage de cartes électroniques qui a besoin d'être reconfigurée d'une façon régulière. Les composants de cette ligne ont été présentés fonctionnellement et organiquement. Ce travail concerne une application de notre laboratoire (IP) à un cas de SPRA. Ainsi, nous avons présenté le projet Agil@Prod dans lequel cette thèse s'inscrit.

Ensuite, nous avons intégré en détail le modèle générique avec le processus de reconfiguration en définissant la structure physique, la structure logique, la configuration physique, la configuration logique, et également les différents types d'opérations pour ce cas d'étude.

Finalement, des manipulations ont été réalisées servant à reconfigurer le système et à déployer une configuration. Ainsi, nous avons créé des interfaces nécessaires, nous avons organisé ensuite une base de données, et également nous avons programmé les automates.

Conclusion générale & Perspectives

L'environnement de production au 21^{ème} siècle fait face à de nombreux défis et changements. L'évolution des marchés est de plus en plus fréquente et imprévisible à cause de la concurrence mondiale. Cette évolution se traduit notamment par l'introduction rapide de nouveaux produits, des changements dans les commandes des clients, de développement des technologies et de l'évolution des réglementations gouvernementales.

En outre, pour qu'une entreprise soit compétitive et rentable, elle doit savoir rester stable dans un contexte international changeant. Elle doit gagner en réactivité et flexibilité et être en mesure de répondre plus rapidement à de nouveaux types de produits, à de nouveaux marchés ainsi qu'à un nouvel environnement sociétal. Par conséquent, les entreprises ont besoin de rationaliser et maîtriser les différents flux d'informations, d'harmoniser leur système d'information et d'améliorer leur rentabilité et leur compétitivité en développant des outils de production capables de s'adapter aux modifications de l'environnement avec une bonne productivité, une haute qualité et un prix bas.

Les travaux présentés dans ce mémoire de thèse ont été réalisés au sein de l'Institut Pascal (UMR 6602 CNRS/UBP/IFMA). Cette thèse avait pour objectif **le développement d'un concept innovant de Systèmes de Production Reconfigurables et Agiles (SPRA)** permettant de s'adapter rapidement et efficacement aux exigences imposées du marché, des clients, de la technologie des procédés, de l'environnement et de la société afin que l'entreprise soit dynamique, compétitive et rentable.

Premièrement, dans la première partie de ce mémoire, nous avons présenté en détail l'évolution des systèmes de production des lignes fixes et dédiées vers des lignes intégrant l'aspect de l'agilité. Ensuite, une classification détaillée des configurations, les outils de modélisation d'un système ainsi que les architectures de pilotage d'un système de production reconfigurable ont été décrits.

Pour répondre à l'objectif de cette thèse, nous avons contribué sur plusieurs aspects du développement d'un concept innovant de SPRA. Ainsi, nous pouvons faire un bilan de nos travaux réalisés en précisant nos apports.

La deuxième partie s'est concentrée sur la phase de modélisation et caractérisation d'un SPRA en utilisant le langage de modélisation des systèmes complexes (SysML). Un modèle générique adapté à ce type de systèmes a été proposé avec une description et une représentation détaillées de chaque composant de ce modèle. Ensuite, nous avons défini les exigences permettant de spécifier les capacités et les contraintes qui doivent être satisfaites par un SPRA, et nous avons également présenté une description de haut niveau de la structure et des fonctionnalités d'un SPRA.

La troisième partie a été consacrée à répondre à la question « comment déploie-t-on une configuration d'un SPRA ? ». Nous avons proposé une démarche à suivre pour concevoir et implanter une nouvelle configuration, et nous avons modélisé les

opérations de reconfiguration sous forme de diagrammes de séquences et de diagrammes d'activité SysML. Egalement, une nouvelle Architecture de Pilotage Hybride Adaptée au SPRA (APHA-SPRA) et un pilotage opérationnel d'un SPRA ont été introduits dans cette partie.

Dans la dernière partie de ce mémoire, nous avons présenté le projet Agil@Prod dans lequel cette thèse s'inscrit. Ainsi, le déploiement de quelques travaux développés au cours de cette thèse a été démontré et validé partiellement sur un démonstrateur industriel au sein de la plate-forme AIP-PRIMECA Auvergne.

En résumé, l'originalité de notre travail a été de modéliser et représenter un système de production en composants réutilisables afin d'assurer la reconfigurabilité et l'agilité de ce système, et de proposer également une manière pratique pour déployer une nouvelle configuration.

En termes de perspectives, les travaux de thèse présentés dans ce mémoire ouvrent de nouvelles voies de recherche, nous pouvons citer :

✚ Répondre à la question « Quand faut-il reconfigurer le système ? ». Nous aurons besoin d'une étude approfondie basée sur plusieurs critères pour que le configurateur puisse prendre la bonne décision.

Les critères pris en compte sont essentiellement le plan de production, les besoins du client, le résultat du diagnostic (en cas de défaillance ou de panne au niveau des ressources), le temps nécessaire à la reconfiguration pour aboutir à la nouvelle configuration, la complexité de sa mise en œuvre, son coût de mise en place, la rentabilité et l'efficacité de la nouvelle configuration, et également l'état actuel du système dans la configuration actuelle (état des produits, mode des ressources). Effectivement, ces critères peuvent guider le configurateur à prendre sa décision importante sur la reconfiguration du système.

✚ Comment passer d'une configuration A à une autre configuration B ? La solution n'est pas toujours triviale. Il faudrait peut être passer par une configuration intermédiaire C par laquelle il faut faire transiter le système pour le positionner dans un état voulu (ou objectif), et cela dépend des états actuels des produits et des ressources en configuration A et également de la structure des configurations A et B.

Des cas de production consistent à démarrer la nouvelle configuration B sans aucun produit, la configuration intermédiaire C sert alors à vider le système de tous les produits en cours de réalisation. Le coût de reconfiguration doit donc prendre en compte le passage éventuel par une configuration intermédiaire.

✚ Comment agencer les ressources en cas de production simultanée de plusieurs produits ?

Chaque configuration est effectivement définie d'une façon à répondre à la gamme logique d'un produit. La configuration est ainsi constituée d'un ensemble de ressources (postes de travail, convoyeurs, robots, logiciels de contrôle, etc.), et

également d'un agencement de ces ressources entre eux (par exemple une structure linéaire, parallèle ou une combinaison de ces deux). Une question importante est alors quelle est la meilleure façon d'organiser et de connecter les différentes ressources pour rendre une production simultanée possible de plusieurs produits ?

✚ Comment prendre en compte les robots mobiles AGV pour créer des systèmes agiles ?

L'utilisation des robots mobiles AGV sert à transporter les pièces en entrée, en sortie du système et aussi entre les postes si c'est nécessaire. En outre, l'AGV avec son convoyeur qui s'interface avec les postes de travail peuvent être reconfigurés pour faire des structures de convoyeurs avec des morphologies différentes, et également les robots AGV sont capables de déplacer d'une manière rapide les différentes ressources dans l'atelier. Cette utilisation multiple des robots mobiles AGV servira alors à faciliter la reconfiguration du système et permettra ainsi de réduire le temps nécessaire pour alimenter le marché avec des produits/services adéquates.

✚ Enfin, comment coupler nos travaux avec les différents travaux sur l'optimisation des layouts des systèmes de production ?

Des travaux d'optimisation pourront intervenir pour aider le configurateur à prendre la bonne décision sur la reconfiguration du système et à choisir également la configuration optimale. Il est à noter que ces travaux serviront ensuite à organiser et à connecter les ressources d'une façon plus efficace et plus économique, et aussi à planifier les différentes utilisations des robots mobiles AGV.

Annexes

Annexe A

Outils et méthodes traditionnels de modélisation d'un système

La modélisation est une manière de décrire le système de sorte que certaines propriétés soient visibles afin d'analyser le système dans le but de le comprendre et de communiquer avec les membres d'une même équipe.

Dans cette annexe, nous allons présenter de façon générale l'évolution d'outils de modélisation d'un système. Nous allons décrire 11 outils ou méthodes traditionnels les plus récurrents dans le milieu industriel et qui continuent d'être utilisés aujourd'hui.

A.1. La méthode APTE

La Méthode APTE (APplication des aux Techniques d'Entreprise) a été créée par Gilbert Barbey en 1964, elle est aujourd'hui déposée par la société APTE. Il s'agit d'une méthode universelle d'analyse fonctionnelle et d'analyse de la valeur pour la conduite de projets d'innovation et d'optimisation. APTE consiste à décrire les besoins sans considérer a priori les solutions, elle permet également d'évaluer l'ensemble des contraintes (techniques, économiques, culturelles...) qui affectent le projet. Donc, elle se base sur l'édition du cahier des charges fonctionnel dès la première phase de conception (Figure A.1). Le but de cette méthode est double, il s'agit d'augmenter la qualité et de réduire le coût de ce que nous étudions.

En outre, APTE définit les problèmes en termes d'objectifs à atteindre et distingue ce qui participe directement aux finalités de ce qui ne dépend que des solutions. Enfin, APTE analyse, ensuite, la qualité et le coût des solutions pour finalement aboutir à la solution optimale.

APTE propose une démarche transversale dans la conduite d'un projet. L'approche fonctionnelle ne concerne pas seulement le produit et les services qu'il rend à son utilisateur, mais également les moyens mis en œuvre pour y parvenir. En d'autres termes, l'analyse fonctionnelle est une démarche qui décrit complètement les fonctions et leurs relations. Elle consiste à rechercher, caractériser, ordonner,

hiérarchiser et valoriser les fonctions. Elle s'intéresse au produit en tant que boîte noire capable de fournir des services dans son environnement durant son cycle d'utilisation. Ainsi, cette méthode APTE s'applique aussi bien aux produits, qu'aux procédés de fabrication, aux équipements et aux organisations.

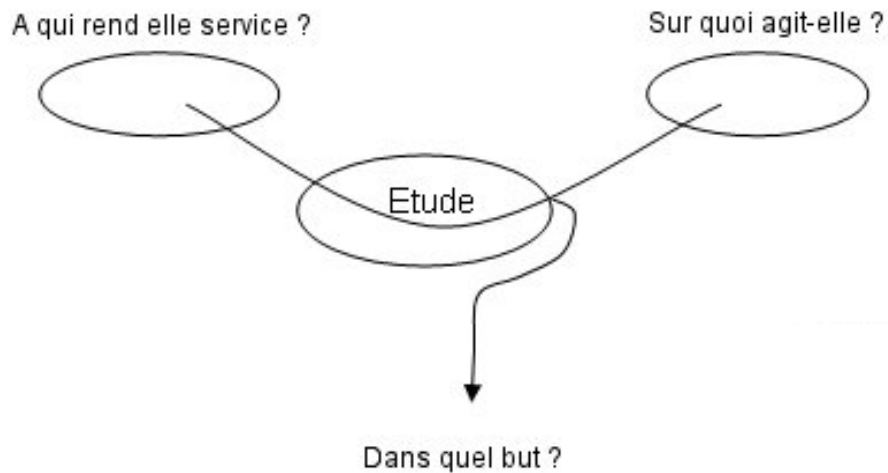


Figure A.1. Outil méthodologique APTE « Bête à cornes ».

Par conséquent, la méthode APTE fait partie des méthodes rapides, simples et efficaces pour répondre aux besoins de modélisation des petites et moyennes entreprises (De la Bretesche, 2000), (APTE, 2013).

A.2. Les méthodes basées sur IDEF

Le langage IDEF (Integrated computer aided manufacturing DEFinition language) a été initié dans les années 1970 au laboratoire des matériaux de la Force Aérienne Américaine (United States Air Force) et complété dans les années 1980 (US Air Force, 1981).

Ce langage a été mis au point en 1993 par la Force Aérienne Américaine et s'est rapidement popularisé avec notamment son utilisation au sein du département de la défense américaine. A partir de ce langage, différentes méthodologies ont été construites pour modéliser différents domaines d'application entourant la spécification d'un système: IDEF0 décomposition des activités (Figure A.2), IDEF1 flux d'informations, IDEF1x structures de données, IDEF2 simulation, IDEF3 description des processus, IDEF4 conception orientée objet, IDEF5 description d'ontologies, IDEF6 rationalités conceptuelles, IDEF7 méthode d'audit pour les systèmes d'information, IDEF8 interfaces utilisateur, IDEF9 conception dirigée par les scénarii des systèmes d'information, IDEF10 architectures d'implantation, IDEF11 artefacts informationnels, IDEF12 organisation du système, IDEF 13

formalisme tri-schémas, IDEF14 conception de réseaux (Mayer *et al.*, 1992), (Kim *et al.*, 2003).

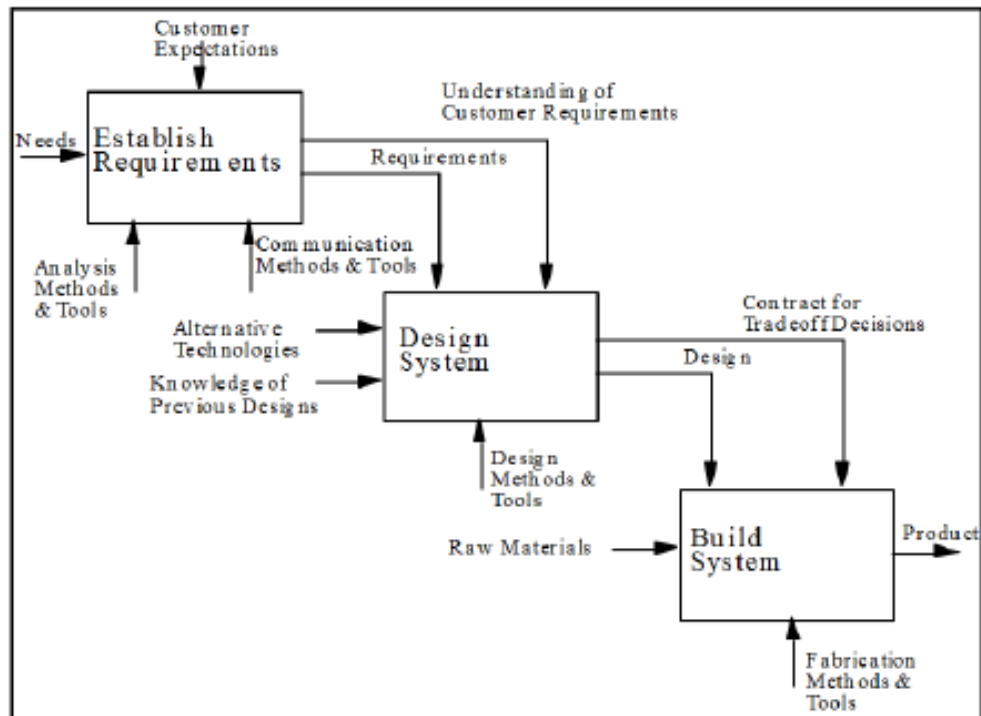


Figure A.2. Modèle IDEF0 : Développement d'un système.

Il est clair que nous avons à notre disposition une multitude de méthodes selon le domaine d'utilisation. En industrie, cette vision constituée de 16 facettes est généralement limitée à l'utilisation des méthodes IDEF0, IDEF1 et IDEF3.

A.3. La méthode MERISE

La méthode MERISE (Méthode d'Étude et de Réalisation Informatique pour les Systèmes d'Entreprise) est une méthode de conception et de développement de système d'information, elle est spécifiquement française et notamment connue comme la concurrente internationale de méthodes anglo-saxonnes telles que SSADM, SDM/S ou Axial. (Rochfeld *et al.*, 1983)

Le but de cette méthode est d'analyser et de concevoir un système d'information. MERISE est basée sur la séparation des données et des traitements à effectuer en plusieurs modèles conceptuels et physiques. En fait, cette séparation assure une durabilité au modèle. Cette méthode se caractérise par deux doubles démarches, par niveau d'abstraction et étapes de décision, et également par des modèles externes vers une vision globale (Figure A.3).

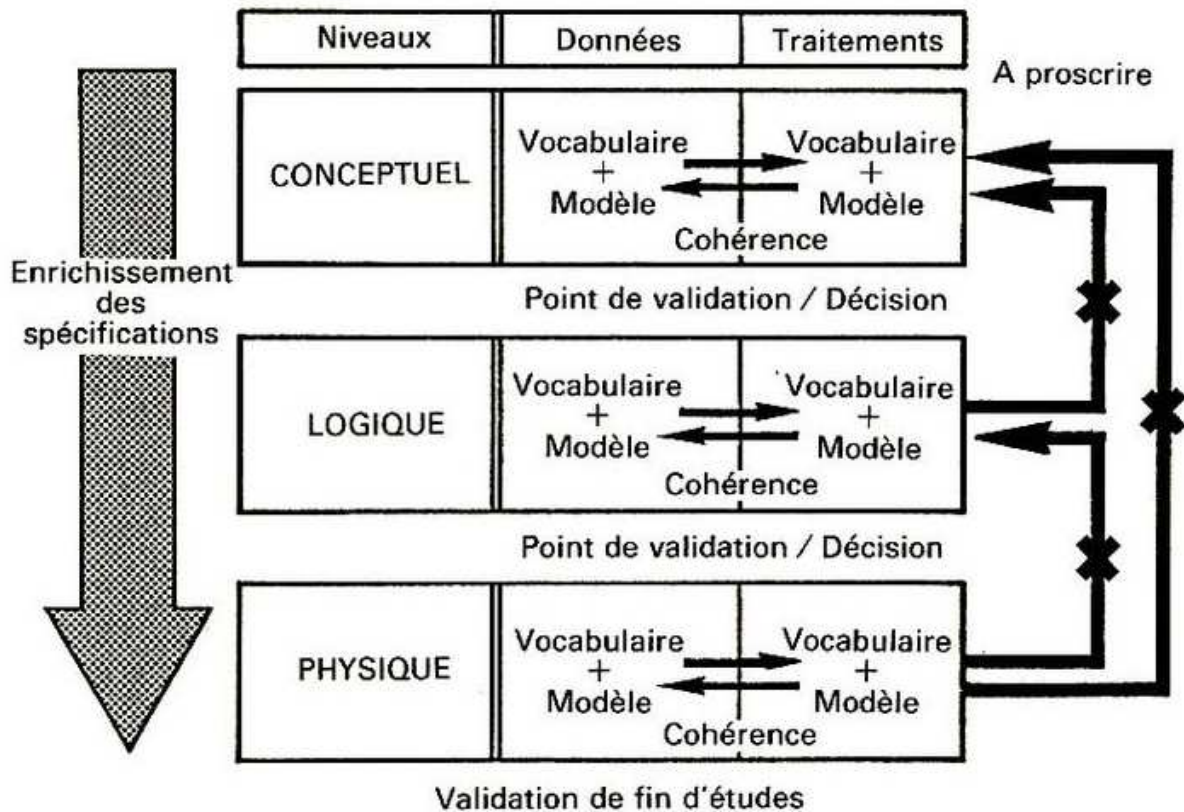


Figure A.3. La méthode MERISE : analyse des données et traitements à chaque niveau.

Une consultation nationale a été lancée en 1977 par le ministère de l'industrie dans le but de choisir des sociétés de conseil en informatique afin de définir une méthode de conception de systèmes d'information. La méthode MERISE a fait suite à cette consultation en 1978-1979. Les deux principales sociétés ayant mis au point cette méthode ont été le CTI (Centre Technique d'Informatique) chargé de gérer le projet, et le CETE (Centre d'Etudes Techniques de l'Équipement) implanté à Aix-en-Provence (Lescourret *et al.*, 1992).

Cette méthode a été très utilisée dans les années 1970 et 1980 pour l'informatisation massive des organisations et avant de voir sa cote diminuée au sein des architectes de systèmes informationnels au profit notamment d'UML.

A.4. La méthode GRAI

Cette méthode GRAI (Graphes à Résultats et Activités Inter-reliés) a été mise au point à partir des années 70. Elle est l'une des principales méthodologies de modélisation d'entreprise, et actuellement reconnue sur le plan international (IFIP et IFAC).

Cette méthode permet en général d'analyser et de concevoir des systèmes décisionnels. Elle a été longtemps utilisée dans les grands groupes manufacturiers

français en amont des spécifications plus techniques. Elle s'appuie sur une description du système en suivant plusieurs approches (Letouzey, 2001), (Dossou, 2003) :

- Le modèle conceptuel de référence GRAI qui décrit le système sous trois angles différents : gestion de production, structure du système de production, et structure décisionnelle du système qui aboutissent à la définition des centres de décision.
- La grille GRAI qui permet de représenter la structure décisionnelle périodique du système de pilotage avec la planification de la production des produits, la gestion des ressources de production mises en œuvre pour le traitement et le transport des produits (Figure A.4).
- Les réseaux GRAI qui reprennent les centres de décision identifiés dans la grille et guident l'utilisateur dans l'identification d'activités, dans le sens de processus de transformation, en les classant en deux catégories : activités de décision et activités d'exécution.

De même que pour MERISE ou IDEF, la méthode GRAI présente une utilisation très pertinente lorsque le contexte du projet que nous avons besoin de modéliser n'a pas vocation à évoluer de façon importante dans le futur ce qui est rare actuellement comme nous l'observons dans toutes les entreprises de production.

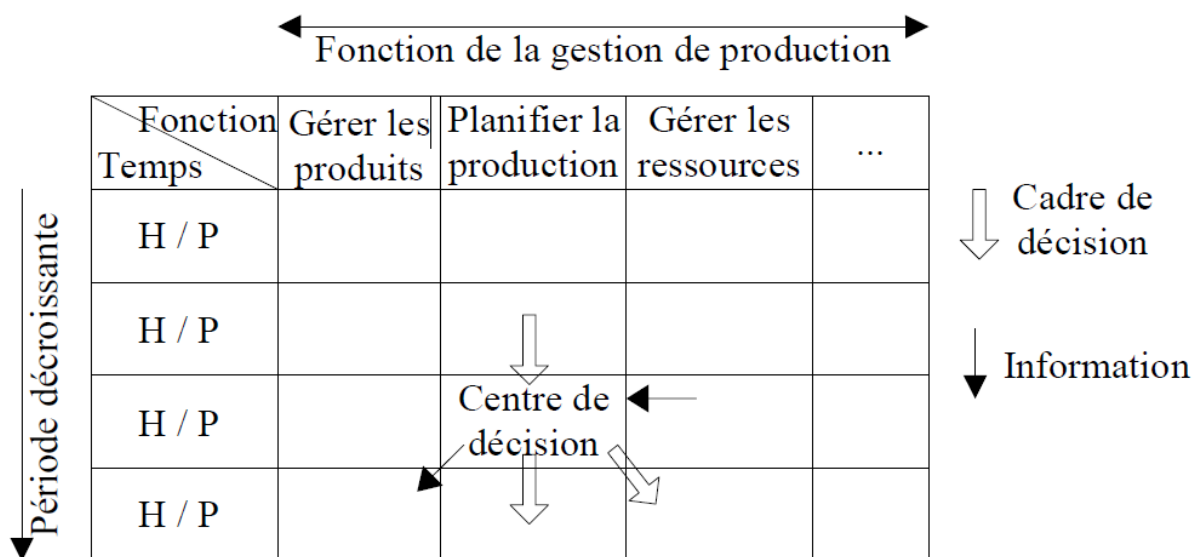


Figure A.4. Grille GRAI.

A.5. L'architecture CIMOSA

Le cube CIMOSA (Computer Integrated Manufacturing – Open System Architecture) compose et décrit les intersections de différents axes de modélisation (Figure A.5) en considérant différentes vues : vue fonctions, vue organisation, vue ressources et vue informations.

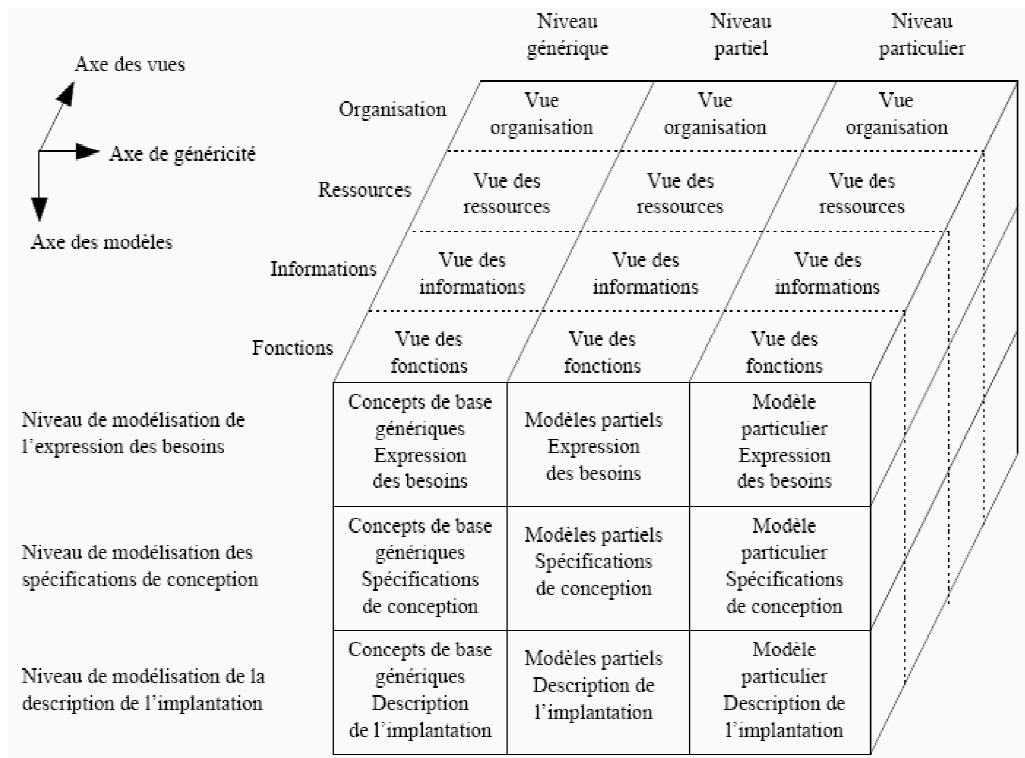


Figure A.5. Matrice des modèles composant CIMOSA.

CIMOSA est une structure entière proposée pour analyser et concevoir les systèmes de production intégrés à grande échelle. Elle a été développée dans la période 1985-1995.

Malheureusement, l'absence de guide pour le formalisme de la description de chaque cellule et le grand nombre de cellules de la matrice rend très difficile sa mise en œuvre qui peut vite devenir inexploitable. (Abdmouleh *et al.*, 2004)

A.6. Les réseaux de Pétri

Depuis leur première définition en 1962 par Carl Adam Pétri dans sa thèse « Communication avec automatismes », les réseaux de Pétri sont devenus un paradigme puissant de modélisation et d'analyse, tant dans le monde industriel que dans les laboratoires de recherche. Ils sont maintenant largement diffusés et de nombreuses études industrielles utilisent cet outil dans un objectif de conception et/ou d'exploitation de systèmes automatisés.

Par définition, un réseau de Pétri est un graphe composé de deux types de nœuds :

- Les places qui permettent de décrire les états du système modélisé.
- Les transitions qui représentent les changements d'état.

Les places et les transitions sont reliées par des arcs orientés (Figure A.6). Une place peut contenir un nombre entier de jetons ou marques. L'ensemble des marques

présentes à un instant donné dans les places constitue le marquage du réseau à cet instant et représente l'état du système. . (Ladet *et al.*, 2009)

Une transition se compose d'un ou de plusieurs arcs d'entrée et d'un ou de plusieurs arcs de sortie. Ce sont ces arcs qui permettront la création de séquences à évolutions parallèles et la description de synchronisations entre ces séquences parallèles. À chaque arc est associé un nombre entier positif appelé « poids de l'arc ». Dans le cas fréquent, où les arcs sont tous de poids 1, nous parlons de réseaux de Pétri ordinaires. Dans le cas contraire, il s'agit de réseaux de Pétri généralisés aussi appelés réseaux « Places-Transitions ».

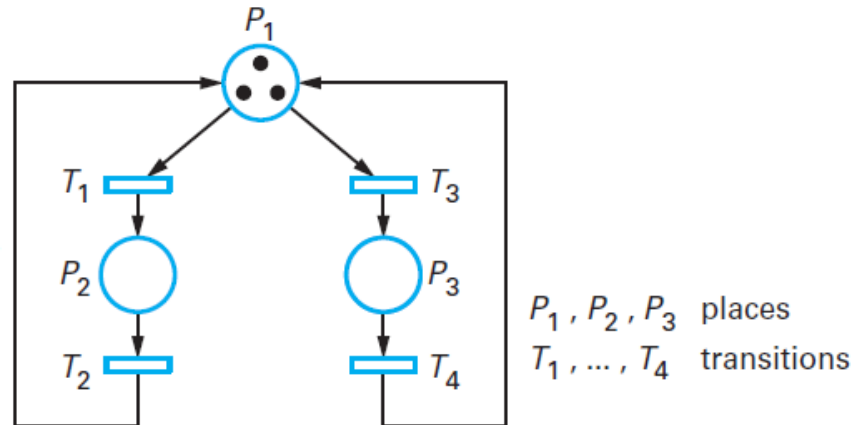


Figure A.6. Exemple de réseau de Pétri.

Les réseaux de Pétri sont donc une représentation graphique permettant de décrire le comportement dynamique d'environnements logiques sous une forme présentant de nombreuses similitudes avec les diagrammes d'états-transitions de type GRAFCET.

Aujourd'hui, avec la complexité croissante de systèmes de production, la conception et l'exploitation des systèmes de production requièrent des modèles plus riches en informations et plus concis que le GRAFCET à des fins d'analyse, de simulation et de commande. La conception d'une installation industrielle peut se résumer en quatre phases : (Techniques de l'ingénieur, 2013)

- La **spécification** des fonctions et de leurs interactions.
- L'analyse et/ou la **validation** de la description obtenue.
- La **simulation** qui complète la connaissance du système projeté et permet un dimensionnement et une évaluation de ses performances.
- L'**exploitation** et la **maintenance**.

Chacune de ces phases repose sur l'utilisation d'un modèle, donc d'un langage. Trop souvent, les outils de modélisation utilisés ne s'appliquent qu'à l'une ou l'autre de ces phases. En plus, le passage d'une étape à la suivante ou le retour en arrière est souvent nécessaire dans cette démarche de conception. Ainsi, la conception de systèmes doit pouvoir s'adapter facilement aux exigences de la production, ce qui

suppose l'utilisation de modèles communs aux différentes étapes de la vie d'une application industrielle. Les réseaux de Pétri se proposent de jouer ce rôle.

L'industrie est maintenant sensibilisée à la démarche de modélisation et d'analyse de systèmes automatisés par l'utilisation des réseaux de Pétri. Elle reste néanmoins le témoin de la complexité toujours croissante des mécanismes séquentiels à décrire, à valider, à simuler et à contrôler.

Il existe de nombreuses variantes et extensions des réseaux de Pétri autonomes qui introduisent les notions de temps, de prédicats ou d'interprétation et de couleurs.

A.7. La méthode SADT

La méthode SADT (Structured Analysis and Design Technics) a été mise au point par la société SofTech aux Etats Unis, développée pour SofTech par Doug Ross en 1977 puis introduite en Europe à partir de 1982 par Michel Galiner, elle se répandit vers la fin des années 1980 comme l'un des standards de description graphique d'un système complexe par analyse fonctionnelle descendante et hiérarchique (**Espinasse, 2003**).

SADT est une méthode d'analyse fonctionnelle et de gestion de projets par niveaux successifs d'approche descriptive de n'importe quel système. Nous pouvons appliquer le SADT à la gestion d'une entreprise tout comme à un système automatisé.

Cette méthode s'intéresse aux fonctions du système et couvre essentiellement la première partie du cycle de vie du système. Les auteurs la présentent comme une méthode pour "communiquer des problèmes". Elle propose une suite cohérente et hiérarchisée de diagrammes (datagrammes et actigrammes), obtenus par raffinements successifs (Figure A.7) : (**Guibert, 2005**)

- Un **datagramme** (diagramme de données) permet de représenter les données par des boîtes et montre les activités qui les créent ou les utilisent par des flèches.

- Un **actigramme** (diagramme d'activités) décrit l'enchaînement des activités, qui sont cette fois représentées par des boîtes, et les données qu'elles manipulent sont représentées par des flèches.

Chaque boîte peut être décomposée en un diagramme plus détaillé. La vérification de chaque étape de décomposition peut se baser sur le respect des informations liées aux flèches où toutes les flèches d'un niveau doivent être conservées au niveau inférieur de décomposition.

En conséquence, la SADT a été une étape vers la description des systèmes informatiques plus complexes mais elle a montré des limites.

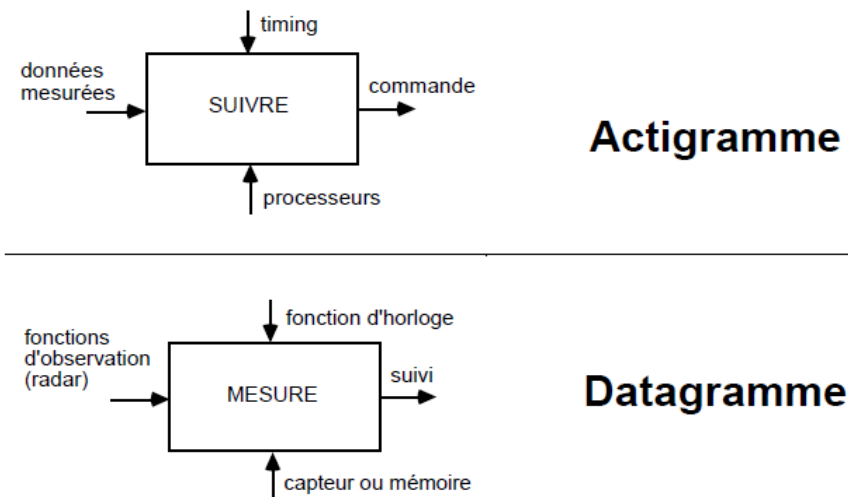


Figure A.7. Exemple : Datagrammes et Actigrammes de SADT.

A.8. La méthode SA-RT

Tout d'abord, SA (Structured Analysis) est une méthode d'analyse qui se base sur la même démarche que SADT comme une suite cohérente et hiérarchisée de diagrammes mais elle s'intéresse plus particulièrement à la partie logicielle du système.

Effectivement, SA-RT (Structured Analysis for Real Time) est un ensemble composé d'outils de l'analyse structurée SA et d'extensions pour le temps réel RT. La méthode SA-RT a été mise au point à la fin des années 80 pour exprimer les spécifications des applications temps réel (Figure A.8). (Cottet *et al.*, 2005)

Le point clef de la méthode SA-RT est la séparation des connaissances selon les trois aspects fondamentaux suivants : (Hladik, 2007)

➡ **L'aspect fonctionnel** (ou transformation de données) : il s'agit d'une représentation de la transformation que le système opère sur les données et d'une spécification des processus qui transforment les données.

➡ **L'aspect dynamique** (pilote par les événements) : c'est une représentation des événements qui conditionnent l'évolution d'un système et une spécification de la logique de contrôle qui produit des actions et des événements en fonction d'événement en entrée et fait changer le système d'état.

➡ **L'aspect informationnel** (données) : il s'agit d'une spécification des données sur les flots ou dans les stockages. Cet aspect, qui est en général assez négligé dans ce type d'application, doit faire l'objet d'une description spécifique.

En général, SA-RT est un outil de communication (formalisme graphique) pour l'expression des besoins logiciels statiques (modèles fonctionnels) et dynamiques (modèles dynamiques). Cet outil fournit un document complet et lisible de

description des besoins du système temps réel pour la conception et la programmation logicielle.

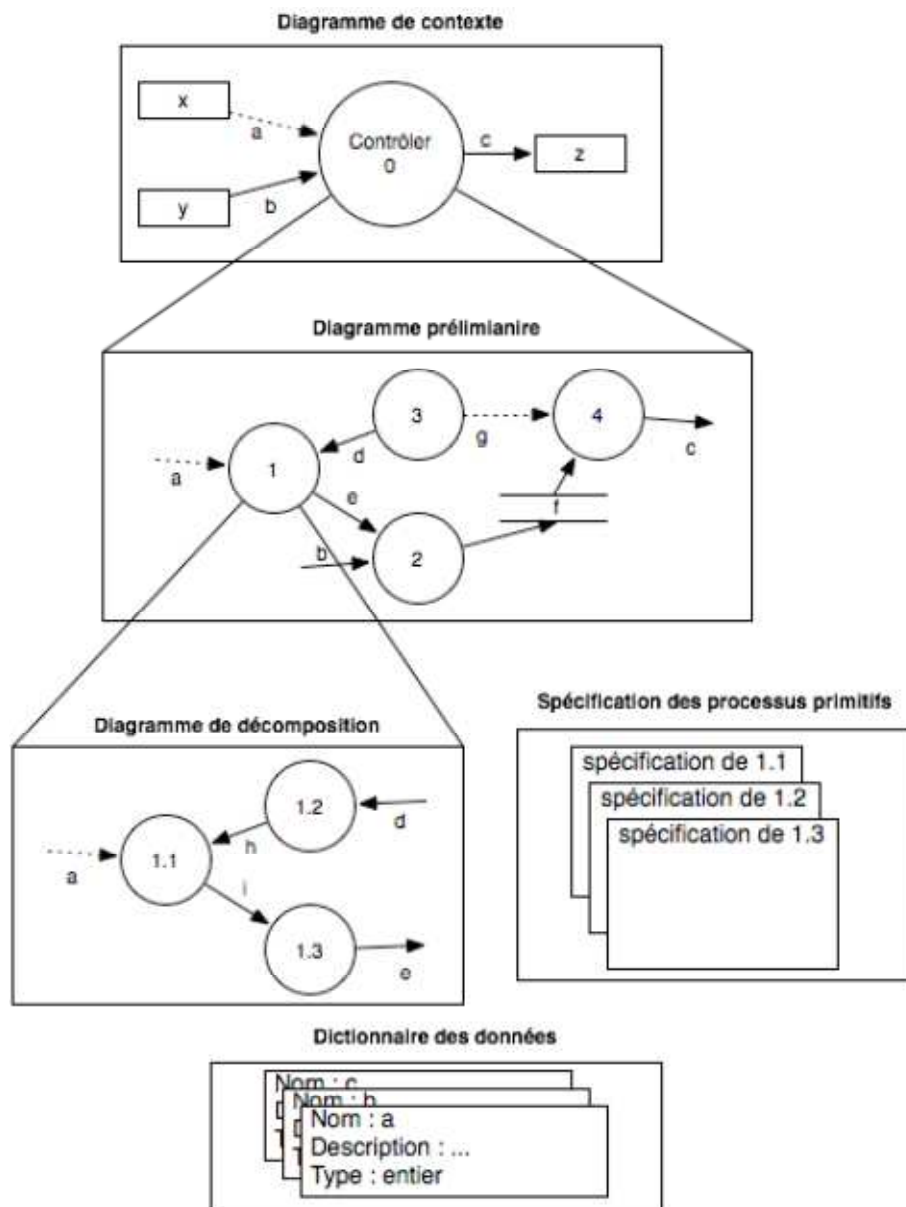


Figure A.8. Organisation générale de la méthode SA-RT.

A.9. La méthode BOOCH

La méthode BOOCH est, avec OMT et OOSE, l'une des méthodes d'analyse et de conception orientées objets à l'origine d'UML. Son nom vient de celui de son concepteur, Grady Booch, qui est aussi l'un des pères du langage UML (avec Ivar Jacobson, fondateur de la méthode OOSE et James Rumbaugh, fondateur de la méthode OMT).

Cette méthode a été publiée en 1991, elle comprend six types de diagrammes comme le diagramme de classes, le diagramme d'objets, le diagramme de transition

d'état, le diagramme du module, le diagramme de processus et le diagramme d'interaction (Figure A.9). (White, 1994)

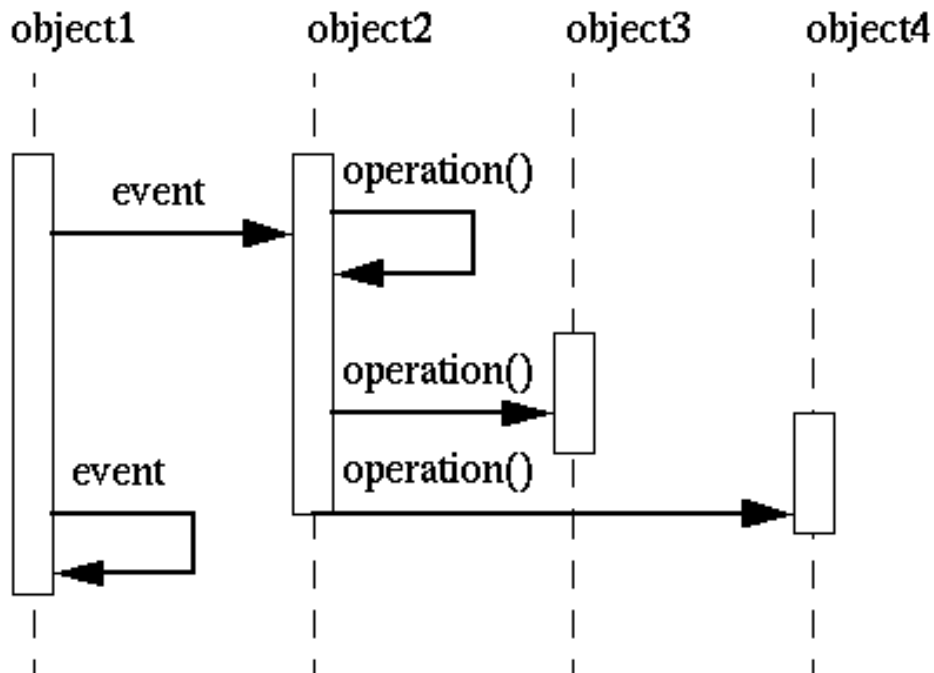


Figure A.9. Diagramme d'interaction de la méthode BOOCH.

L'objectif de cette méthode est de faciliter l'implémentation de programmes dans des langages de programmation orientée objet, ainsi que de représenter les différentes phases du développement d'un projet. Pour les programmeurs, l'avantage est d'obtenir une modélisation d'une solution qui soit indépendante de son implémentation dans un langage particulier (par exemple : Java, C++,...).

A.10. La méthode OMT

La méthode OMT (Object Modeling Technique) est un langage de modélisation et conception de logiciels. Elle est développée en 1991 par James Rumbaugh comme une méthode pour développer des systèmes orientés objets et pour soutenir la programmation orientée objet (Derr, 1995).

En outre, OMT a trois diagrammes principaux : diagramme d'objet (Figure A.10), diagramme dynamique et diagramme fonctionnel.

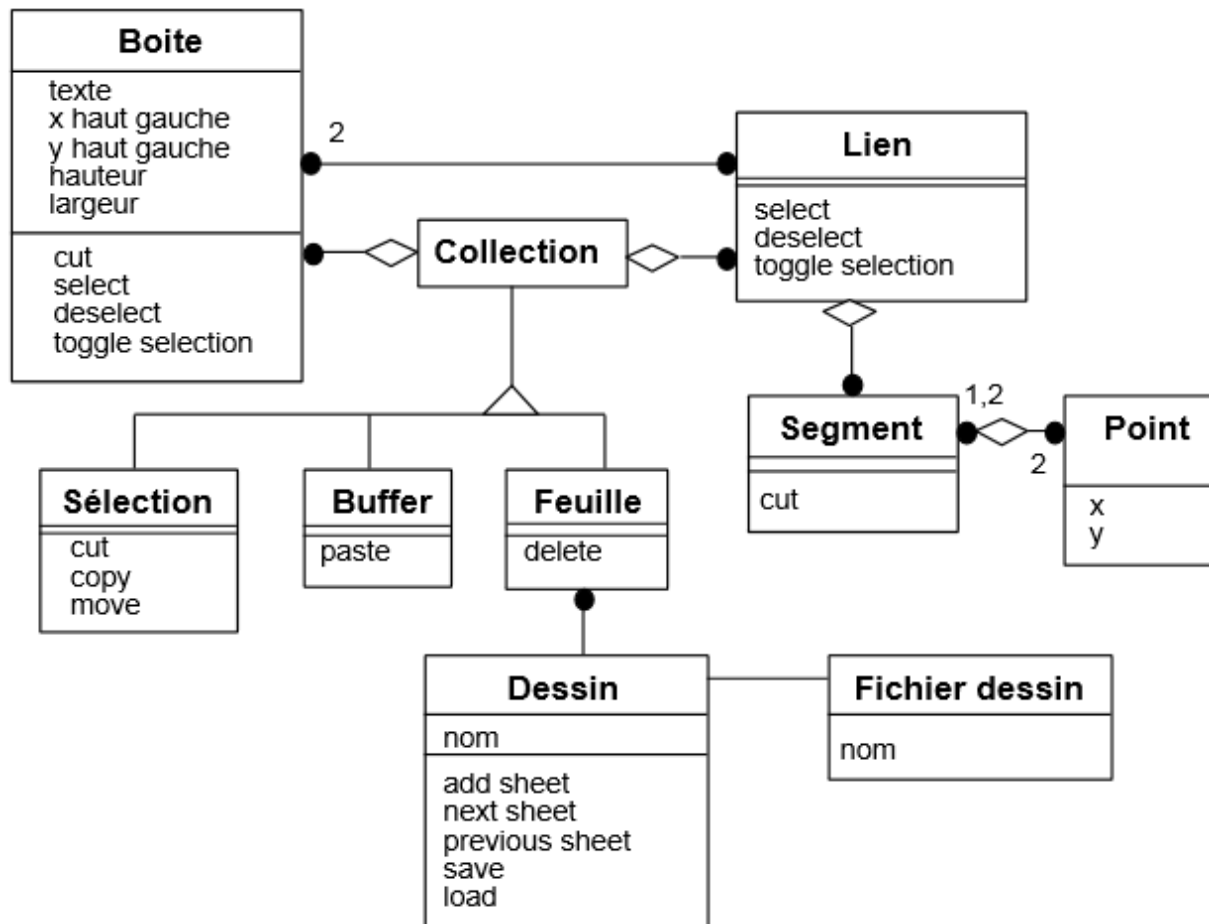


Figure A.10. Diagramme d'objet de la méthode OMT.

A.11. La méthode OOSE

Cette méthode OOSE (Object Oriented Software Engineering) est une technique de conception de logiciels qui est utilisée dans la conception de logiciels en programmation orientée objet.

OOSE est développée par Ivar Jacobson en 1992, c'est la première méthodologie de conception orientée objet qui emploie des cas d'utilisation dans la conception de logiciels. La méthode OOSE est basée sur les 5 modèles suivants (Figure A.11) : (Wirsing *et al.*, 2009)

- ❖ **Le modèle d'analyse des besoins** : c'est un modèle qui se situe au niveau de l'interaction entre l'individu et l'environnement. Ce modèle est également appelé le modèle d'analyse de cas d'utilisation.

- ❖ **Le modèle d'analyse** : il permet de définir les différentes relations entre les objets, les acteurs et le système.

- ❖ **Le modèle de conception** : il s'agit d'une modélisation de ces relations à travers différents diagrammes.

- ❖ **Le modèle d'implémentation** : ce modèle permet le passage au code.
- ❖ **Le modèle de test** : il est basé sur les essais d'actions et d'interactions du système.

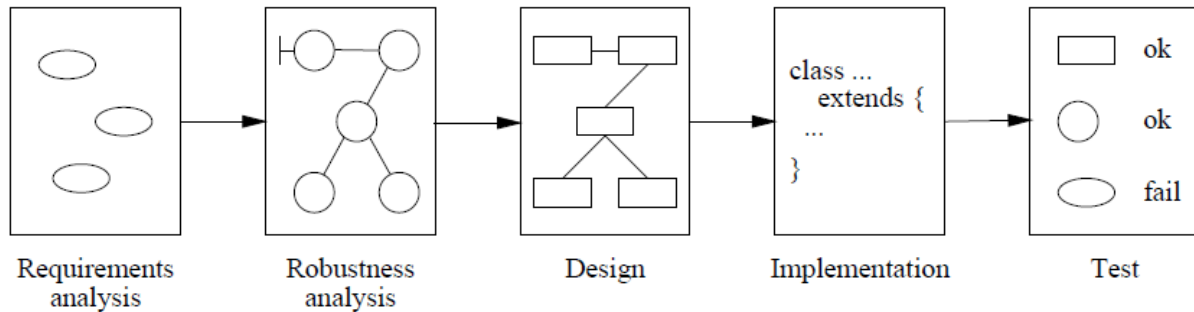


Figure A.11. Phases de développement de la méthode OOSE.

A.12. Conclusion et discussion

Cette annexe a permis de décrire les différentes méthodes traditionnelles de modélisation d'un système. Ces méthodes et leurs représentations sont très pratiques et permettent de mettre en œuvre rapidement les premières phases de la spécification. Elles conduisent également à une implémentation rapide d'un système dans le cadre d'un développement spécifique.

Nous pouvons les classer en méthodes orientées comportement qui s'intéressent à la dynamique du système (ex : réseaux de Pétri), méthodes fonctionnelles qui s'intéressent aux fonctions du système (ex : SADT), méthodes orientées données qui ne s'intéressent pas aux traitements (ex : MERISE) et méthodes orientées objets qui ne séparent pas les données des traitements (ex : BOOCH, OMT). (Anvar *et al.*, 2006)

Ces méthodes d'analyse et de conception traditionnelles ont été élaborées dans des contextes spécifiques, elles sont donc très efficaces lorsqu'elles sont utilisées dans leur environnement originel mais montrent rapidement leurs limites pour d'autres domaines d'utilisation. Chaque méthode a son propre langage, et les personnes chargées de la spécification amont d'un projet utilisent en général un éventail de méthodes selon la phase de conception dans laquelle ils se trouvent et selon le contexte technologique associé. Il est donc parfois difficile de s'y retrouver et de mettre en place des liens cohérents entre les différentes spécifications.

De plus, ce sont toutes d'excellentes méthodes d'analyse mais elles ne permettent pas de construire une modélisation qui peut s'adapter aux changements du contexte de réalisation ou de l'évolution des besoins au cours du temps, ce à quoi peuvent répondre les méthodes orientées objets (Mertins *et al.*, 1997).

Par conséquent, il est clair que la modélisation du système est une condition nécessaire à la réussite d'un projet notamment lors des phases de spécification. Nous avons abordé un ensemble d'outils méthodologiques permettant d'analyser et de guider les concepteurs tout au long du processus de spécification d'un système, mais ils ne s'intègrent pas au sein d'une vraie démarche cohérente de modélisation d'un système évolué dans sa globalité, ce que pourrait permettre le langage UML ou SysML.

Annexe B

Diagrammes d'exigences d'un SPRA

Spécifier un SPRA impose de définir et d'analyser l'ensemble des exigences tant au niveau système qu'au niveau de l'ensemble de ses constituants.

Ainsi, nous allons relier chacun des acteurs principaux aux exigences correspondantes afin d'obtenir des demandes simples, et d'affecter et de lier ces exigences à des éléments ciblés du modèle. Cette annexe récapitule ces différents diagrammes développés en utilisant Artisan Studio.

B.1. Diagramme des exigences du configurateur

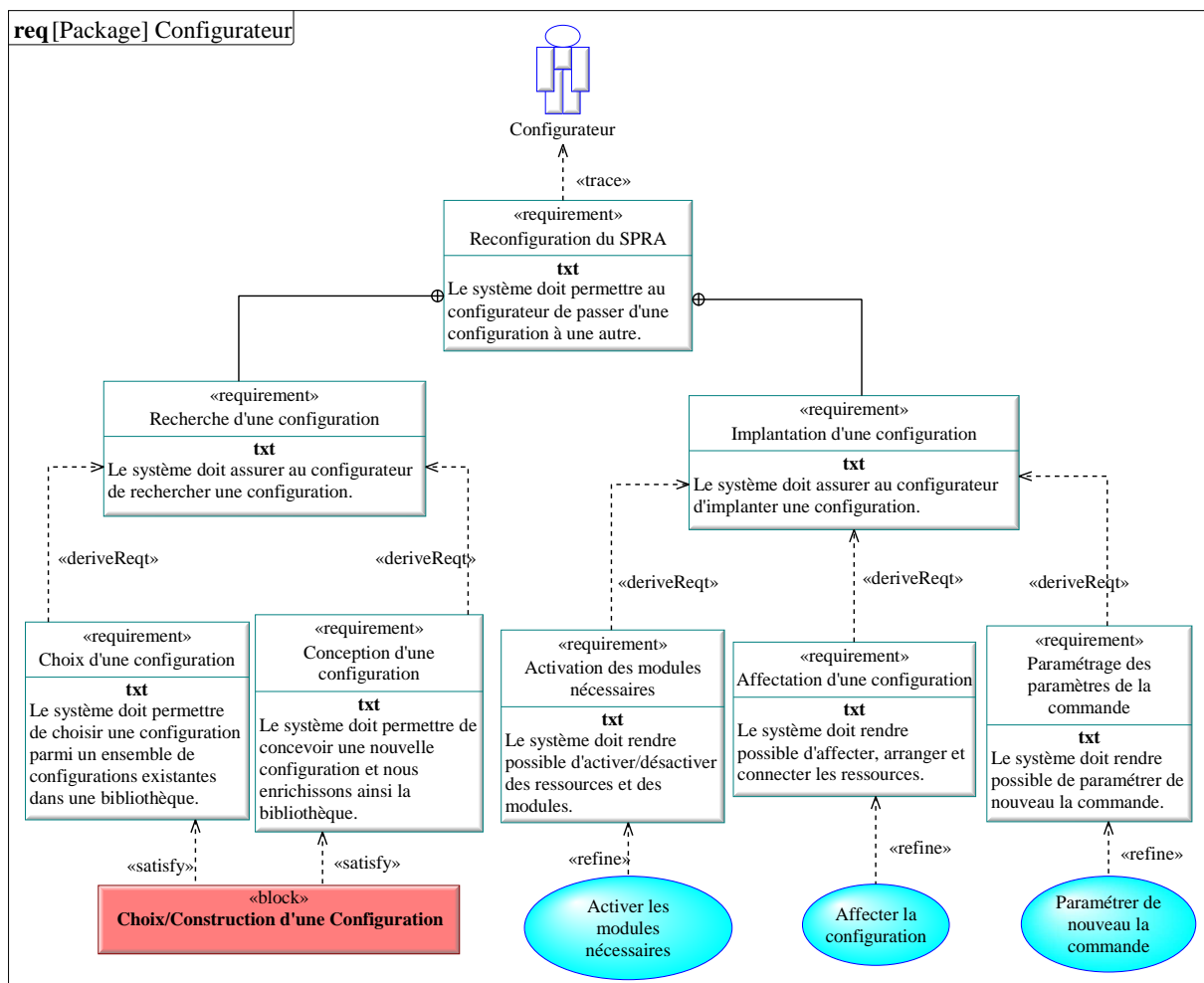


Figure B.1. Diagramme des exigences du configurateur.

B.2. Diagramme des exigences du chef de l'atelier

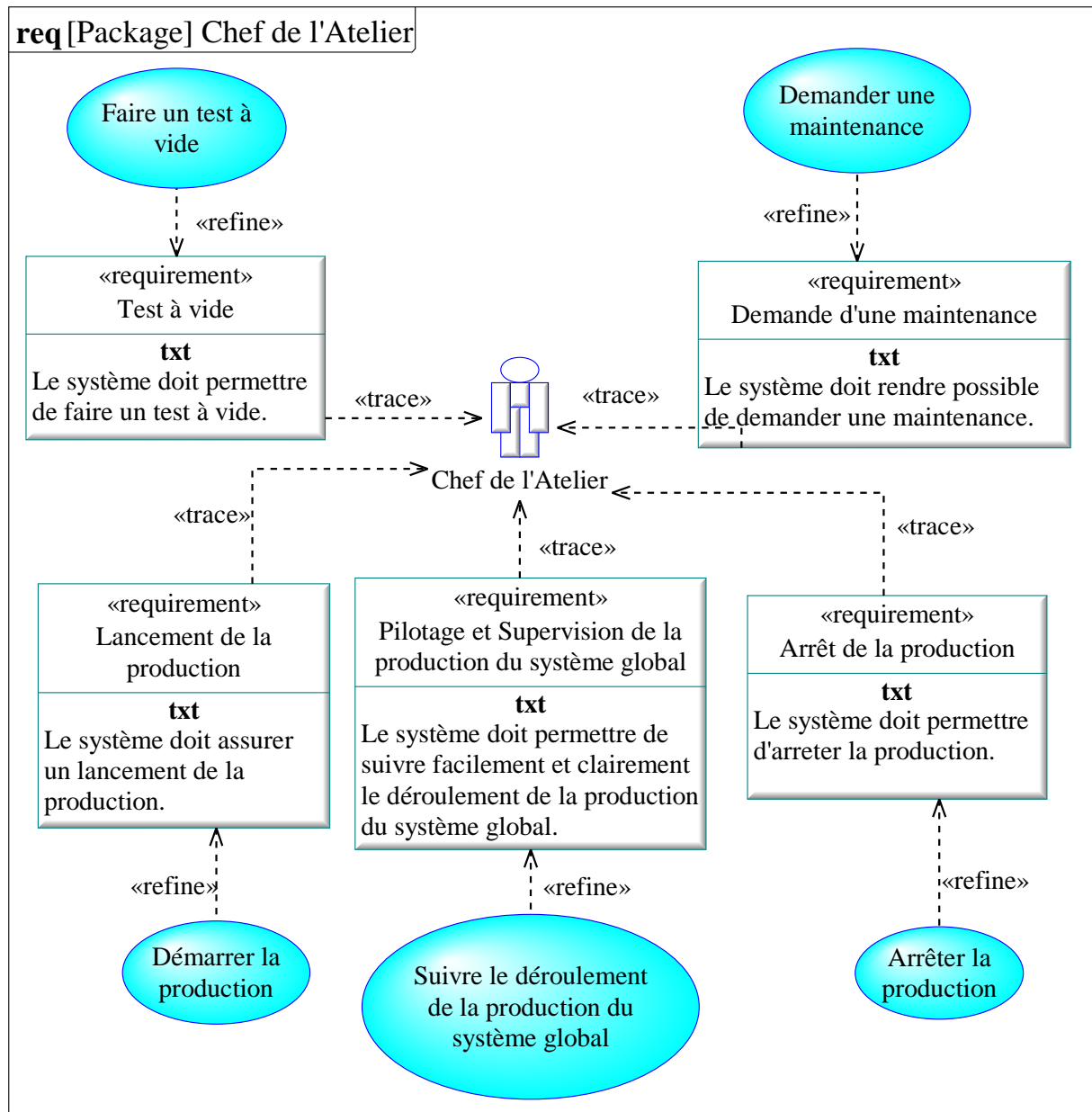


Figure B.2. Diagramme des exigences du chef de l'atelier.

B.3. Diagramme des exigences d'opérateurs de maintenance

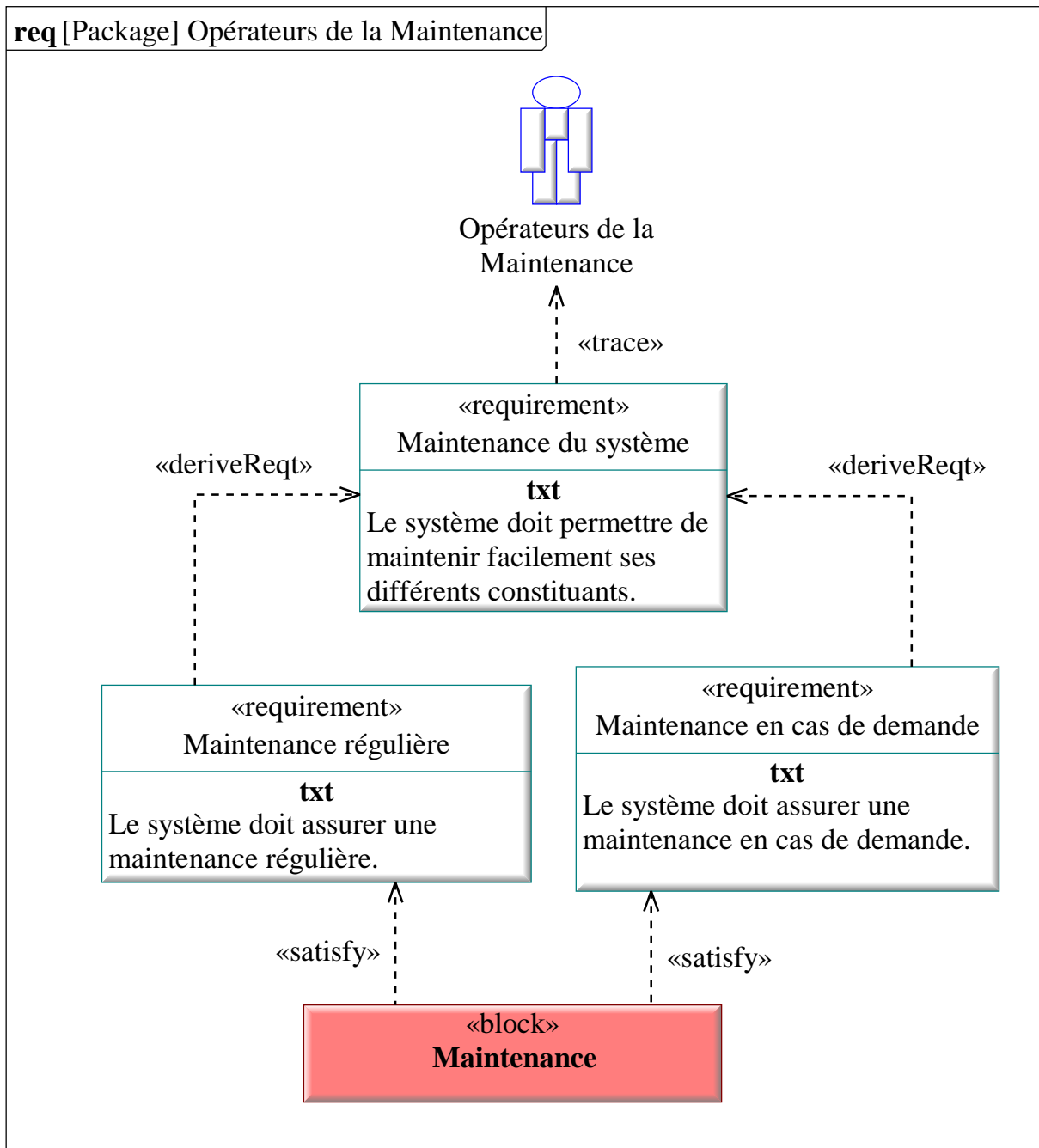


Figure B.3. Diagramme des exigences d'opérateurs de maintenance.

B.4. Diagramme des exigences d'opérateurs

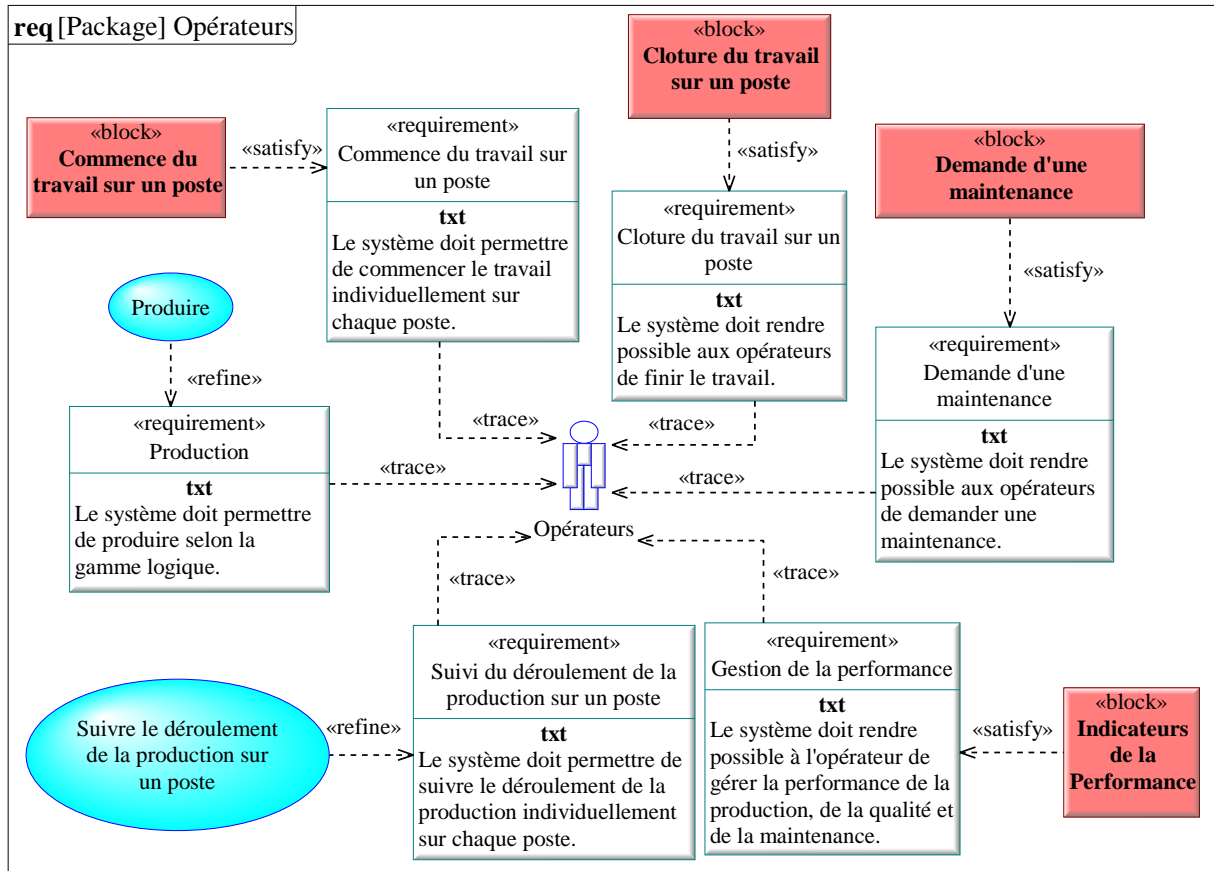


Figure B.4. Diagramme des exigences d'opérateurs.

Annexe C

Diagrammes de cas d'utilisation opérationnels d'un SPRA

Pour tout projet d'ingénierie complexe comme notre système SPRA, le diagramme de cas d'utilisation est susceptible de contenir un grand nombre de cas d'utilisation, ce qui rend très difficile la compréhension et le contexte du projet.

Une meilleure approche consiste à construire des diagrammes de cas d'utilisation opérationnels pour les acteurs principaux et cela est généralement suffisant. Cette annexe récapitule ces différents diagrammes développés en utilisant Artisan Studio.

C.1. Diagramme de cas d'utilisation d'un SPRA

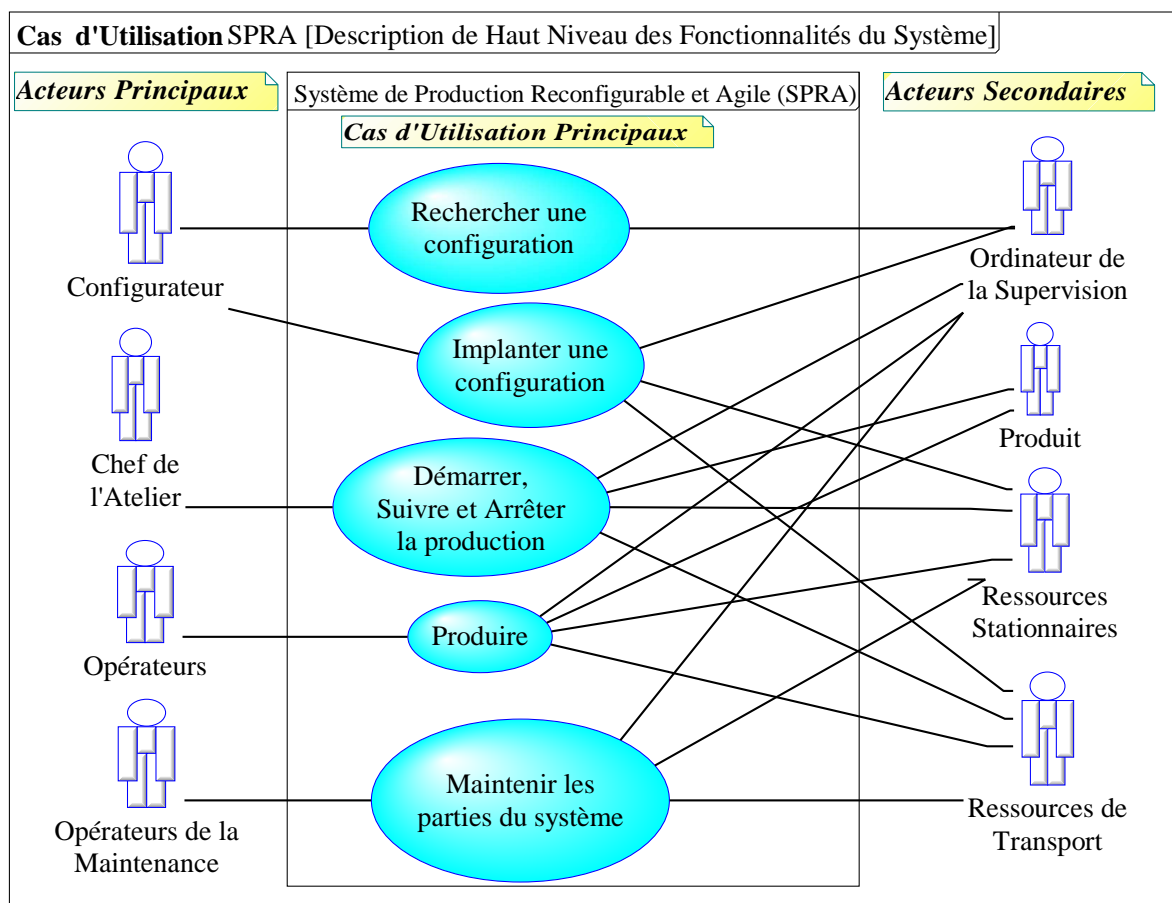


Figure C.1. Diagramme de cas d'utilisation d'un SPRA.

C.2. Diagramme de cas d'utilisation opérationnels du configurateur

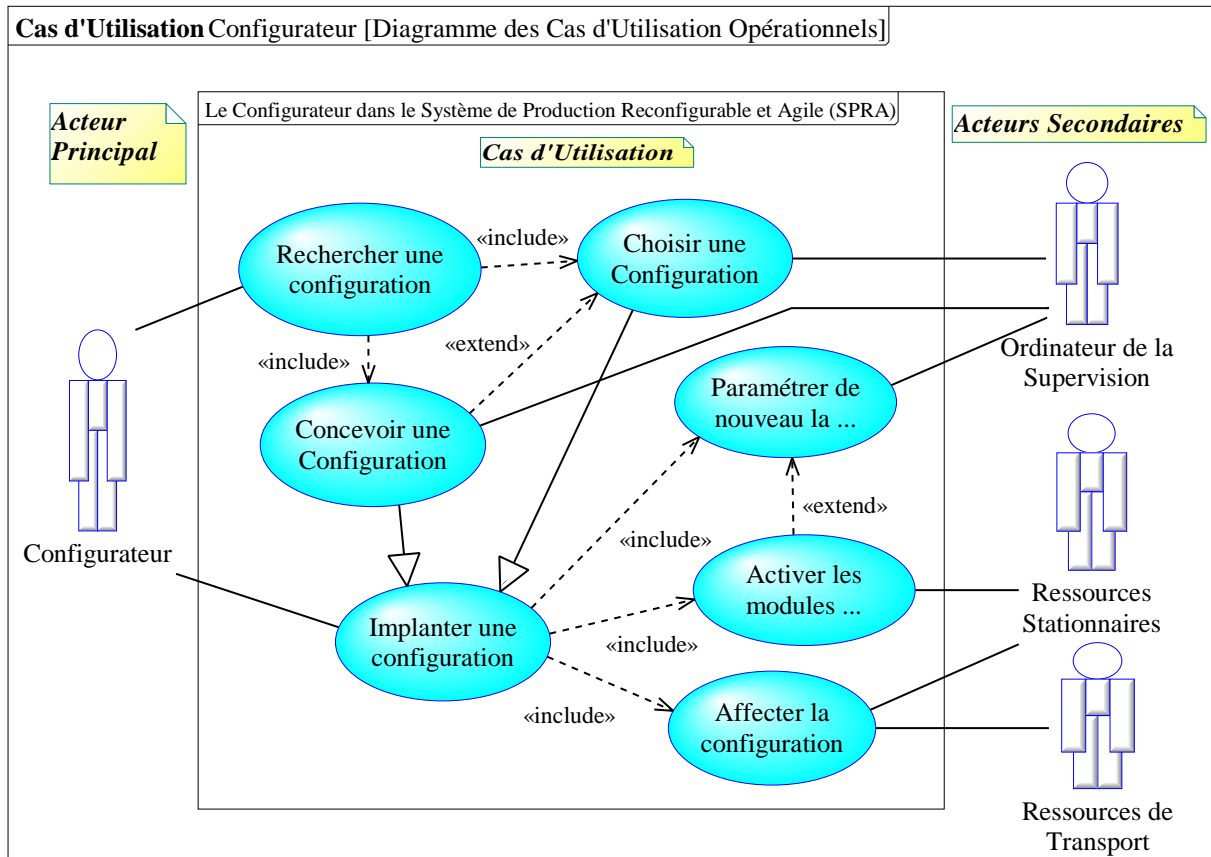


Figure C.2. Diagramme de cas d'utilisation opérationnels du configurateur.

C.3. Diagramme de cas d'utilisation opérationnels du chef de l'atelier

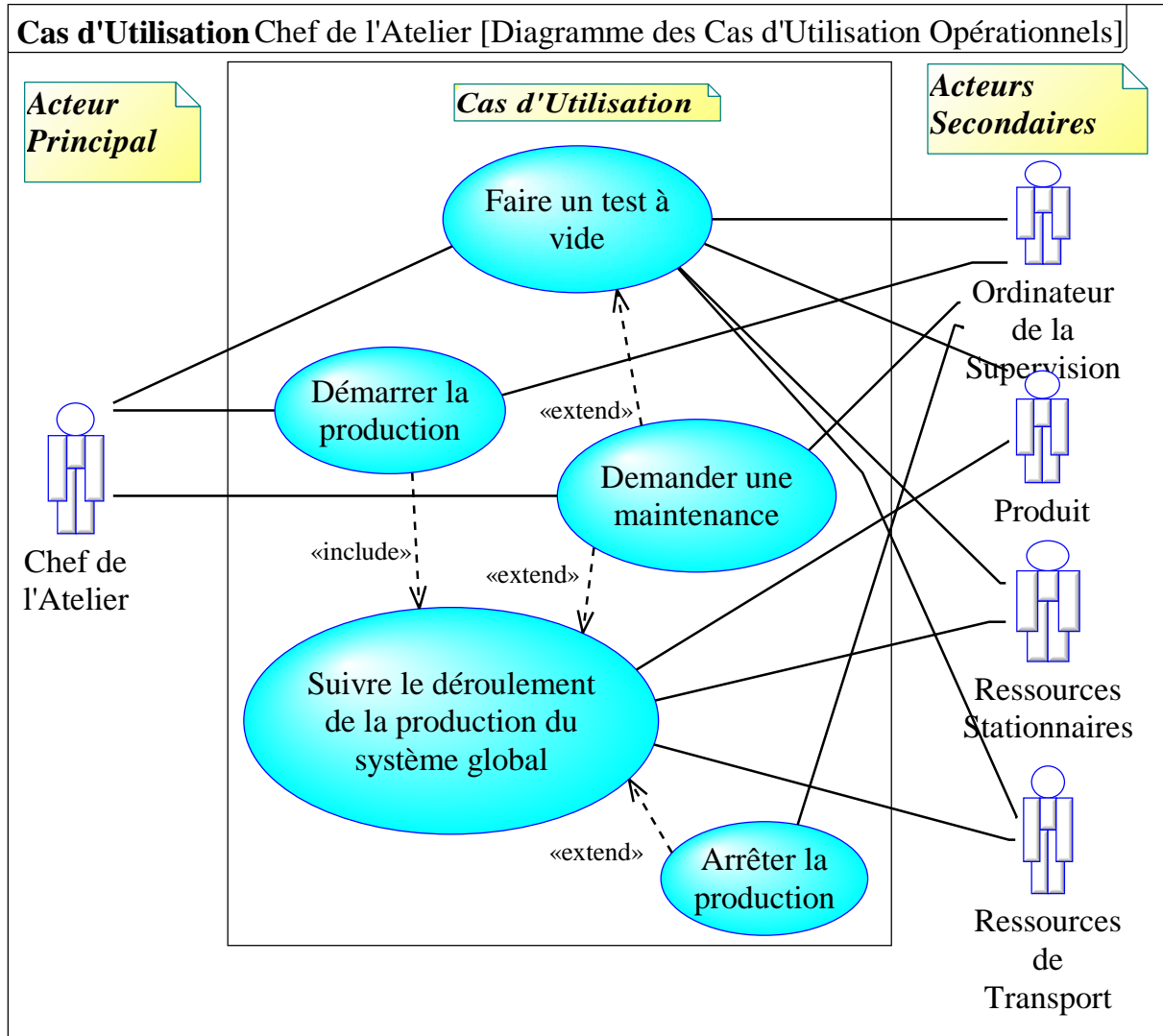


Figure C.3. Diagramme de cas d'utilisation opérationnels du chef de l'atelier.

C.4. Diagramme de cas d'utilisation opérationnels d'opérateurs

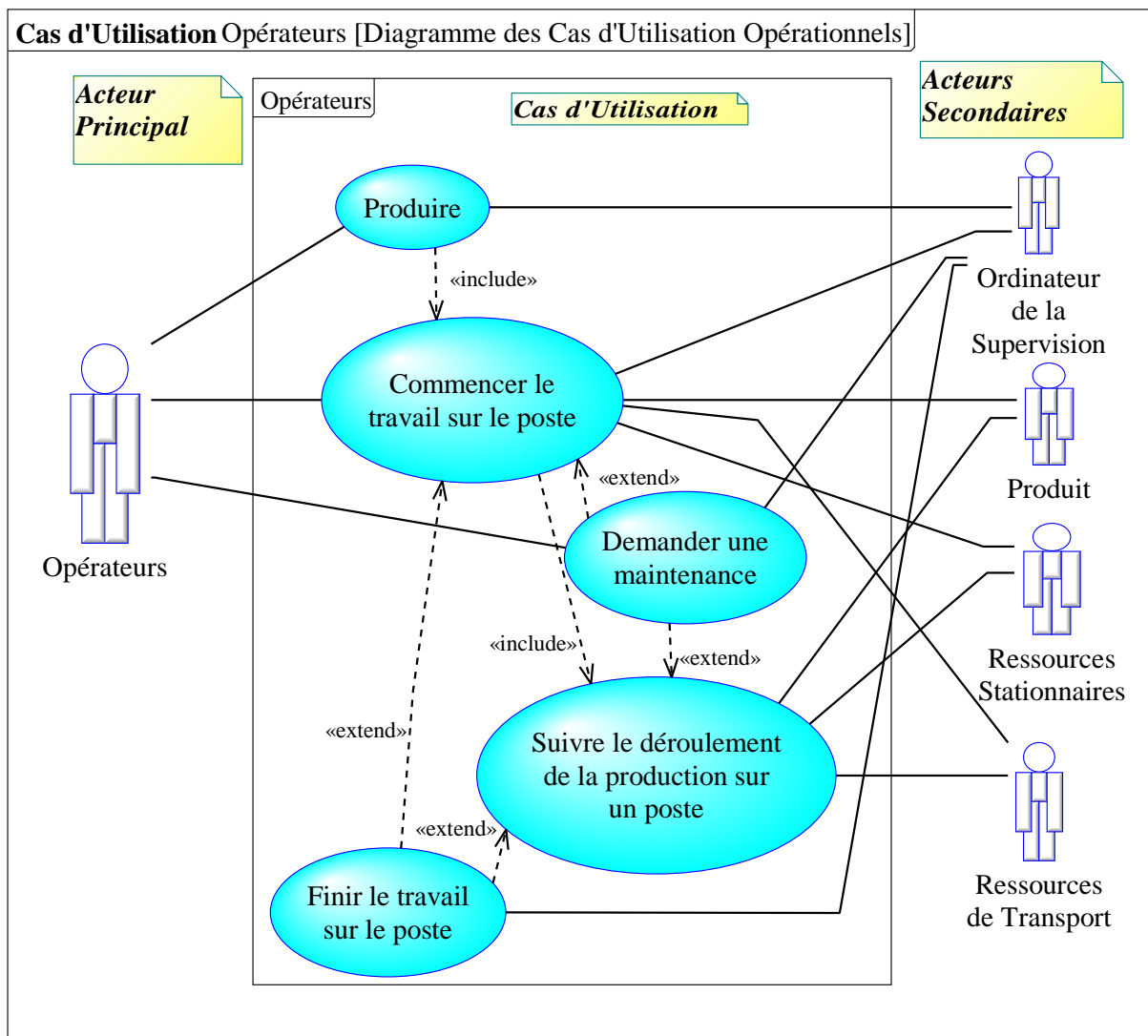


Figure C.4. Diagramme de cas d'utilisation opérationnels d'opérateurs.

C.5. Diagramme de cas d'utilisation opérationnels d'opérateurs de la maintenance

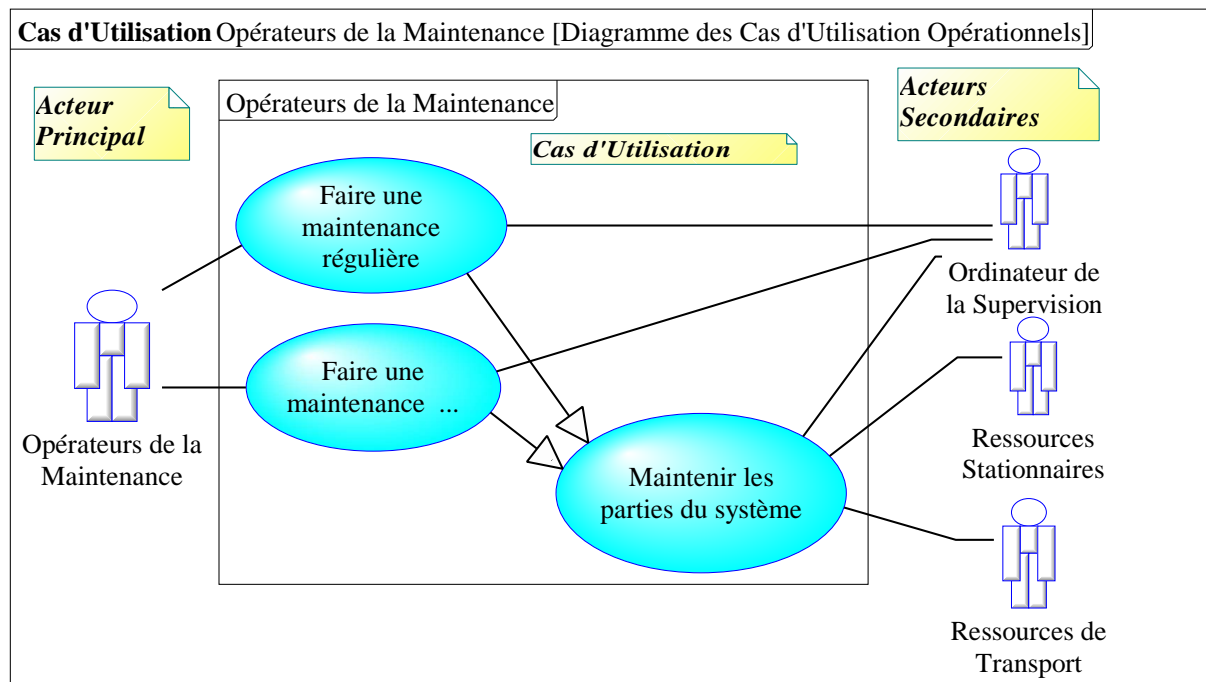


Figure C.5. Diagramme de cas d'utilisation opérationnels d'opérateurs de la maintenance.

Références

- (**Abdmouleh et al., 2004**) Abdmouleh A., Spadoni M., Vernadat F., « Distributed client/server architecture for CIMOSA-based enterprise components », *Computers in Industry*, vol. 55, 2004, p. 239 - 253.
- (**Adhau et al., 2012**) Adhau S., Mittal M. L., Mittal A., « Multi-agent system for distributed multi-project scheduling: An auction-based negotiation approach », *Engineering Applications of Artificial Intelligence*, vol. 25, 2012, p. 1738-1751.
- (**AFIS, 2009**) Association Française d'Ingénierie Système (AFIS), « Découvrir et comprendre l'Ingénierie Système », Ouvrage collectif AFIS préparé par le Groupe de Travail Ingénierie Système, Paris, décembre 2009.
- (**AFIS, 2013**) AFIS, www.afis.fr, mai 2013.
- (**Ali, 2009**) Ali A., « L'approche multi-agents pour le pilotage des systèmes complexes : Appliquée aux Systèmes du Trafic Urbain », Thèse de doctorat, Université Blaise Pascal – Clermont II, 2009.
- (**Andersson, 1997**) Andersson N., « On Modelling and Implementing Shop floor Control Systems », PhD Thesis, Chalmers University of technology, ISBN 91-7197-450-4, 1997.
- (**Anvar et al., 2006**) Anvar S., Terrier F., Gérard S., « Introduction à UML », Laboratoire d'Intégration des Systèmes et des Technologies (LIST), CEA/Saclay, 2006.
- (**APTE, 2013**) APTE, Génie méthodologique, <http://www.methode-apte.com>, mai 2013.
- (**Arentsen, 1995**) Arentsen A. L., « A Generic Architecture for Factory Activity Control », PhD Thesis, University of Twente, The Netherlands, 1995.
- (**Atego, 2010**) Atego, Artisan Studio, « SysML Tutorial », <http://www.Atego.com>, Version 7.2-a, 2010.
- (**Aube, 2013**) Aube Electronique «Cartes Electroniques», <http://www.aube-electronique.com>, Septembre 2013.
- (**Audibert, 2009**) Audibert L., « UML 2: De l'apprentissage à la pratique », Editions Ellipses, 2009.
- (**Barkmeyer et al., 1999**) Barkmeyer E., Denno P., Feng S., Jones A., Wallace E., « NIST Response to MES Request for Information », NISTIR 6397, National Institute of Standards and Technology, Gaithersburg, MD, 1999.
- (**Berruet, 2007**) Berruet P., « Simulation of reconfigurable systems: from control code simulation to reflective simulation », *Proceedings 21st European Conference on Modelling and Simulation (ECMS)*, 2007.

- (Bi et al., 2007)** Bi Z., Lang S., Shen W., « Reconfigurable manufacturing systems: the state of the art », *International Journal of Production Research*, vol. 46, n° 4, 2007, p. 967-992.
- (Blanc et al., 2008)** Blanc P., Demongodin I., Castagna P., « A holonic approach for manufacturing execution system design: An industrial application », *Engineering Applications of Artificial Intelligence*, vol. 21, 2008, p. 315-330.
- (Booch et al., 2000)** Booch G., Rumbaugh J., Jacobson I., « Le guide de l'utilisateur UML », Eyrolles, 2000.
- (Booch et al., 2007)** Booch G., Maksimchuk R. A., Engel M. W., Young B. J., Conallen J., Houston K. A., « Object-Oriented Analysis and Design with Applications », 3rd Edition, Pearson Education, 2007.
- (Boss et al., 2010)** Boss J., Dove R., « Agile Aircraft Installation Architecture in a Quick Reaction Capability Environment », *INCOSE International Symposium*, Chicago, IL, USA, 12-15 July 2010.
- (Boursbia et al., 2002)** Boursbia S., Trentesaux D., « Self-Organization in Distributed Manufacturing Control: state-of-the-art and future trends », *IEEE-SMC*, Hammamet-Tunisie, 2002.
- (Bruccoleri et al., 2003)** Bruccoleri M., Amico M., Perrone G., « Distributed intelligent control of exceptions in reconfigurable manufacturing systems », *International Journal of Production Research*, vol 41, n° 7, 2003, p. 1393-1412.
- (Cardin, 2007)** Cardin O., « Apport de la simulation en ligne dans l'aide à la décision pour le pilotage des systèmes de production, application à un système flexible de production », Thèse de doctorat, Université de Nantes, 2007.
- (Cardin et al., 2009)** Cardin O., Castagna P., « Using online simulation in Holonic manufacturing systems », *Engineering Applications of Artificial Intelligence*, vol. 22, 2009, p. 1025-1033.
- (Carr et al., 2004)** Carr C. T., McGinnity T. M., McDaid L. J., « Integration of UML and VHDL-AMS for analogue system modeling », *Formal Aspects of computing*, vol. 16, Issue 1, 2004, p. 80-94.
- (Chalfoun et al., 2011)** Chalfoun I., Kouiss K., Bouton N., Ray P., « Des lignes fixes et dédiées vers l'agilité des systèmes, quelques outils de modélisation », 7ème Conférence Internationale en Conception et Production Intégrée (CPI'2011), Oujda-Maroc, Octobre 2011.
- (Chalfoun et al., 2012a)** Chalfoun I., Kouiss K., Bouton N., Ray P., « Characterization of a Reconfigurable and Agile Manufacturing System (RAMS) », The 14th International Conference on Modern Information Technology in the

- Innovation Processes of the Industrial Enterprises (MITIP 2012), Budapest-Hungary, October 2012.
- (Chalfoun et al., 2012b)** Chalfoun I., Kouiss K., Bouton N., Ray P., « Analysis, modeling and design of a Reconfigurable and Agile Manufacturing System (RAMS) », "Robotica & Management" International Journal, Vol. 17, No. 2, December 2012.
- (Chalfoun et al., 2013)** Chalfoun I., Kouiss K., Huyet A-L, Bouton N., Ray P., « Proposal of a generic model for the Reconfigurable and Agile Manufacturing Systems (RAMS) », The 46th CIRP Conference on Manufacturing Systems (CMS 2013), Sesimbra-Portugal, May 2013.
- (Chalfoun et al., 2014)** Chalfoun I., Kouiss K., Bouton N., Ray P., « Specification of a Reconfigurable and Agile Manufacturing System (RAMS) », un article a été soumis le 24 Juillet 2014 au " International Journal of Mechanical Engineering and Automation (IJMEA) ".
- (Chiron, 2009)** Chiron F., « Contribution à la flexibilité et à la rapidité de conception des systèmes automatisés avec l'utilisation d'UML », Thèse de doctorat, Université Blaise Pascal – Clermont II, 2009.
- (Cottet et al., 2005)** Cottet F., Grolleau E., « Systèmes temps réel de contrôle-commande, conception et implémentation », Dunod, 2005.
- (Cunha et al., 2007)** Cunha P. F., Maropoulos P. G., « Digital Enterprise Technology: Perspectives and Future Challenges », Springer, 2007.
- (Darmoul, 2010)** Darmoul S., « Etude de la contribution des systèmes immunitaires artificiels au pilotage de systèmes de production en environnement perturbé », Thèse de doctorat, Université Blaise Pascal – Clermont II, 2010.
- (Dashchenko, 2006)** Dashchenko A. I., « Reconfigurable Manufacturing Systems and Transformable Factories », Netherlands, Editions Springer, 2006.
- (David, 2009)** David P., « Contribution à l'analyse de sûreté de fonctionnement des systèmes complexes en phase de conception : application à l'évaluation des missions d'un réseau de capteurs de présence humaine », Thèse de doctorat, Université d'Orléans, 2009.
- (De la Bretesche, 2000)** De la Bretesche B., « La méthode APTE: Analyse de la Valeur et Analyse Fonctionnelle », Editions Petrelle, 2000.
- (Derr, 1995)** Derr K. W., « Applying OMT: A Practical Step-by-Step Guide to Using the Object Modeling Technique », SIGS Books, New York, 1995.
- (Dilts et al., 1991)** Dilts D. M., Boyd N. P., Whorms H. H., « The Evolution of Control Architectures for Automated Manufacturing Systems », Journal of Manufacturing Systems, vol 10, n° 1, 1991, p. 79-93.

- (Dobre, 2010)** Dobre D., « Contribution à la modélisation d'un système interactif d'aide à la conduite d'un procédé industriel », Thèse de doctorat, Université Henri Poincaré, Nancy 1, 2010.
- (Dossou, 2003)** Dossou P. E. M., « Modélisation des raisonnements pour l'aide à la conception des systèmes de production dans un environnement GRAI », Thèse de doctorat, Université de Bordeaux I, 2003.
- (Doumeingts *et al.*, 1995)** Doumeingts G., Vallespir B., Chen D., « Methodologies for designing CIM systems: A survey », *Computers in Industry*, vol. 25, 1995, p. 263 - 280.
- (Dove, 2001)** Dove R., « Response Ability: The Language, Structure, and Culture of Agile Enterprise », John Wiley & Sons, 2001.
- (Dove, 2013a)** Dove R., « Agile Systems and Processes: Necessary and Sufficient Fundamental Architecture (Agile 101) », INCOSE Webinar, 11 January 2013.
- (Dove, 2013b)** Dove R., « Agile Systems & Systems Engineering », International Workshop, Jacksonville, FL, USA, 28-29 Jan 2013.
- (Dudra, 2003)** Dudra S., « Object-Oriented Design Approach to Systems Engineering of a Mechanical Steering System », SAE Technical Paper, International Truck & Bus Meeting & Exhibition, Ft. Worth, Texas, USA, 2003.
- (ElMaraghy, 2006)** ElMaraghy H. A., « Flexible and reconfigurable manufacturing systems paradigms », *International Journal of Flexible Manufacturing Systems*, vol 17, 2006, p. 261-276.
- (ElMaraghy, 2011)** ElMaraghy H. A., « Enabling Manufacturing Competitiveness and Economic Sustainability », *Proceedings of the 4th International Conference on Changeable, Agile, Reconfigurable and Virtual production (CARV 2011)*, Montreal, Canada, 2-5 Octobre 2011.
- (ElMaraghy, 2012)** ElMaraghy W., « Managing Complexity in Engineering Design & Manufacturing », Séminaire à l'IFMA, Clermont-Ferrand, 13 Décembre 2012.
- (Endres *et al.*, 2003)** Endres A., Rombach D., « A Handbook of Software and Systems Engineering, Empirical Observations, Laws and Theories », *The Fraunhofer IESE Series on Software Engineering*, Pearson Addison Wesley ISBN 0 321 15420 7, May 2003.
- (Espinasse, 2003)** Espinasse B., « Méthodes fonctionnelles : SADT », Professeur à l'université d'Aix-Marseille, 2003.
- (Friedenthal *et al.*, 2003)** Friedenthal S., Burkhart R., « Extending UML from Software to Systems », INCOSE Symposium, July 2003.

- (Friedenthal *et al.*, 2012)** Friedenthal S., Moore A., Steiner R., « A Practical Guide to SysML: The Systems Modeling Language », Elsevier, USA, 2012.
- (Guibert, 2005)** Guibert O., « SADT : System Analysis and Design Technic », Cours d'Analyse et Conception des Systèmes d'Information, IUT de l'Université Bordeaux 1, avril 2005.
- (Gullander, 1999)** Gullander P., « On Reference Architecture for Development of Flexible Cell Control Systems », PhD Thesis, Chalmers University of Technology, Sweden, 1999.
- (Guo *et al.*, 2009)** Guo Q., Zhang M., « A novel approach for multi-agent-based Intelligent Manufacturing System », Information Sciences, vol. 179, 2009, p. 3079-3090.
- (Hafri *et al.*, 2001)** Hafri Y., Najid N. M., « Utilisation de l'approche multi-agents pour le pilotage en temps réel des systèmes de production », Conférence Francophone de Modélisation et Simulation " Conception, Analyse et Gestion des Systèmes Industriels " (MOSIM'01), Troyes-France, 2001.
- (Hause, 2006)** Hause M., « The SysML Modelling Language », Proceedings of the 5th European Systems Engineering Conference, Edimbourg, 18-20 septembre 2006.
- (Hladik, 2007)** Hladik P. E., « Document de synthèse sur les méthodologies SART », INSA Toulouse, 2007.
- (Holt *et al.*, 2008)** Holt J., Perry S., « SysML for Systems Engineering », The Institution of Engineering and Technology, London, United Kingdom, 2008.
- (Huang, 2002)** Huang C. Y., « Distributed manufacturing execution systems: A workflow perspective », Journal of Intelligent Manufacturing, vol. 13, 2002, p. 485-497.
- (IEEE 1220, 1999)** The Standard IEEE 1220, « Standard for application and Management of the Systems Engineering Process », 1999.
- (INCOSE, 2004)** International Council on Systems Engineering (INCOSE), Technical Board International Council on Systems Engineering, « Systems engineering Handbook », INCOSE-TP-2003-016-02, Version 2a, 1 June 2004.
- (ISA-S95, 2004)** ANSI/ISA-S95, « Enterprise - Control System Integration, Part 3: Activity Models of Manufacturing Operations Management », USA, 2004.
- (Joergensen *et al.*, 2011)** Joergensen S. N., Madsen O., Nielsen K., Joergensen K. A., « Design Requirements For Designing Responsive Modular Manufacturing Systems », The 21st International Conference on Production Research « Innovation in Product and Production » ICPR 21, Germany, 2011.

- (Kim *et al.*, 2003) Kim C. H., Weston R. H., Hodgson A., Lee K. H., « The complementary use of IDEF and UML modelling approaches », *Computers in Industry*, vol. 50, 2003, p. 35-56.
- (Koestler, 1967) Koestler A., « The Ghost in the Machine », Editions Arkana, Londres, 1967.
- (Koren *et al.*, 1999) Koren Y., Heisel U., Jovane F., Moriwaki T., Pritschow G., Ulsoy G., Van Brussel H., « Reconfigurable Manufacturing Systems », *Annals of the CIRP*, vol. 48, n° 2, 1999, p. 527-540.
- (Koren, 2010a) Koren, Y., « The Global Manufacturing Revolution: Product-Process-Business Integration and Reconfigurable Systems », John Wiley & Sons, 2010.
- (Koren *et al.*, 2010b) Koren, Y., Shpitalni M., « Design of reconfigurable manufacturing systems », *Journal of Manufacturing Systems*, Technical paper, vol. 29, 2010, p. 130-141.
- (Kouiss *et al.*, 2003) Kouiss K., Pezet N., « Pilotage d'un système de production à l'aide d'une simulation en ligne », CPI 2003, Meknès-Maroc, 2003.
- (Ladet *et al.*, 2009) Ladet P., DEMONGODIN I., « Outils de modélisation des automatismes séquentiels : Réseaux de Petri », Professeurs à l'Institut National Polytechnique de Grenoble et l'Université Paul-Cézanne, Aix-Marseille, 2009.
- (Lamotte *et al.*, 2005) Lamotte F., Berruet P., Philippe J. L., « A model for the reconfiguration of manufacturing systems », The 16th IFAC World Congress Proceedings, Prague, 2005.
- (Lamotte, 2006) Lamotte F., « Proposition d'une approche haut niveau pour la conception, l'analyse et l'implantation des systèmes reconfigurables », PhD Thesis, Université de Bretagne Sud, 2006.
- (Le Moigne, 1999) Le Moigne J. L., « La modélisation des systèmes complexes », Dunod, Paris, 1999.
- (Le Moigne, 2006) Le Moigne J. L., La théorie du système en général : Théorie de la modélisation, e-book, Collection les classiques du réseau intelligence de la complexité, France, 2006.
- (Lee, 2006) Lee E. J., « Reconfiguration dynamique de la commande d'un système manufacturier : approche par la synthèse de la commande », Thèse de doctorat, Ecole Centrale de Lille, 2006.
- (Leitao *et al.*, 2008) Leitao P., Restivo F., « A holonic approach to dynamic manufacturing scheduling », *Robotics and Computer-Integrated Manufacturing*, vol. 24, 2008, p. 625–634.

- (Leitao, 2009)** Leitao P., « Agent-based distributed manufacturing control: A state-of-the-art survey », *Engineering Applications of Artificial Intelligence*, vol. 22, 2009, p. 979 - 991.
- (Lenclud, 1993)** Lenclud T., « Contribution à la conception d'un système intégré de simulation des systèmes de production », Thèse de doctorat, Université de Valenciennes et du Hainaut-Cambrésis, 1993.
- (Lescourret *et al.*, 1992)** Lescourret F., Pérochon L., Coulon J. B., Faye B., Landais E., « Modelling an information system using the MERISE method for agricultural research: The example of a database for a study on performances in dairy cows », *Agricultural Systems*, vol. 38, Issue 2, 1992, p. 149 - 173.
- (Letouzey, 2001)** Letouzey A., « Ordonnancement interactif basé sur des indicateurs: Applications à la gestion de commandes incertaines et à l'affectation des opérateurs », Thèse de doctorant de l'Institut National Polytechnique de Toulouse, Laboratoire Génie de Production, 2001.
- (Malhotra *et al.*, 2009)** Malhotra V., Raj T., Arora A., « Reconfigurable manufacturing system: an overview », *International Journal of Machine Intelligence*, vol. 1, 2009, p. 38-46.
- (Malhotra *et al.*, 2010)** Malhotra V., Raj T., Arora A., « Excellent Techniques of Manufacturing Systems: RMS and FMS », *International Journal of Engineering Science and Technology*, vol. 2, n° 3, 2010, p. 137-142.
- (Mayer *et al.*, 1992)** Mayer R. J., Michael C., deWitte P. S., « IDEF Family of Methods for Concurrent Engineering and Business Re-engineering Applications », *Knowledge Based Systems*, 1992.
- (McFarlane *et al.*, 2003)** McFarlane D., Sarma S., Chirn J. L., Wong C. Y., Ashton K., « Auto ID systems and intelligent manufacturing control », *Engineering Applications of Artificial Intelligence*, vol. 16, 2003, p. 365-376.
- (McFarlane *et al.*, 2013)** McFarlane D., Giannikas V., Wong C. Y., Harrison M., « Product intelligence in industrial control: Theory and practice », *Annual Reviews in Control*, vol. 37, 2013, p. 69–88.
- (Mehrab *et al.*, 2000a)** Mehrabi M. G., Ulsoy A. G., Koren Y., « Reconfigurable Manufacturing Systems: key to future manufacturing », *Journal of Intelligent Manufacturing*, vol. 11, n° 4, 2000, p. 403-419.
- (Mehrab *et al.*, 2000b)** Mehrabi M. G., Ulsoy A. G., Koren Y., « Reconfigurable manufacturing systems and their enabling technologies », *Int. J. Manufacturing Technology and Management*, vol. 1, n° 1, 2000, p. 114–131.

- (**Mehrabi et al., 2002**) Mehrabi M.G., Ulsoy A.G., Koren Y., Heytler P., « Trends and perspectives in flexible and reconfigurable manufacturing systems », *Journal of Intelligent Manufacturing*, vol. 13, 2002, p. 135-146.
- (**Meinadier, 1998**) Meinadier J. P., « Ingénierie et intégration des systèmes », Editions Hermès, Paris, 1998.
- (**Meinadier, 2002**) Meinadier J. P., « Le métier d'intégration de systèmes », Hermès Science Publications, Edition Lavoisier, 2002.
- (**Mendez, 2002**) Mendez H., « Synthèse des lois de surveillance pour les procédés industriels complexes », Thèse de doctorat, Institut National Polytechnique de Grenoble, Grenoble, 2002.
- (**Mertins et al., 1997**) Mertins K., Jochem R. Jäkel F. W., « A tool for object-oriented modelling and analysis of business processes », *Computers in Industry*, vol. 33, 1997, p. 345 - 356.
- (**Meyer et al., 2009**) Meyer G. G., Framling K., Holmstrom J., « Intelligent Products: A survey », *Computers in Industry*, vol. 60, 2009, p. 137-148.
- (**Monassier, 2009**) Monassier P., « Informatique Industrielle : Supervision et IHM », formation CESI, Université Lyon 1, 2009.
- (**Morel et al., 2003**) Morel G., Panetto H., Zaremba M., Mayer F., « Manufacturing enterprise control and management system engineering: paradigms and open issues », *Annual Reviews in Control*, vol. 27, 2003, p. 199-209.
- (**Mouchard, 2002**) Mouchard J. S., « Proposition d'une approche méthodique pour la conception des systèmes automatisés de production, application aux systèmes transiques », Thèse de doctorat, Université de Bretagne Sud, Lorient, 2002.
- (**Nagel, 1992**) Nagel R. N., « 21st Century Manufacturing Enterprise Strategy Report », AMEF N0001-92 Prepared for the Office of Naval Research, Arlington, Virginia, 1992.
- (**NASA, 1995**) National Aeronautics and Space Administration (NASA), « Systems Engineering Handbook: Fundamentals of Systems Engineering », SP-610S, June 1995.
- (**Nguyen et al., 2004**) Nguyen K. D., Zhenxin S., Thiagarajan P. S., Weng-Fai W., « Model-driven design via executable UML to SystemC », *Real-Time Systems Symposium, Proceedings of the 25th IEEE International*, 5-8 Dec. 2004, p. 459 - 468.
- (**OMG, 2007**) OMG: Object Management Group, Systems Modeling Language (OMG SysML), version 1.0, 2007.

- (OMG, 2008)** OMG: Object Management Group, Systems Modeling Language (OMG SysML), version 1.1, 2008.
- (OMG, 2009)** OMG: Object Management Group - Unified Modeling Language, OMG Specification-UML 2.2 Superstructure & UML 2.2 Infrastructure, 2009.
- (OMG, 2010)** OMG: Object Management Group, Systems Modeling Language (OMG SysML), version 1.2, 2010.
- (OMG, 2011)** OMG: Object Management Group, Unified Modeling Language (OMG UML), Infrastructure, version 2.4, 2011.
- (OMG, 2013)** OMG: Object Management Group, <http://www.omg.org>, 2013.
- (Pach et al., 2011)** Pach C., Zambrano G., Berger T., Sallez Y., Trentesaux D., Deneux D., « Maitrise de la myopie des systèmes flexibles de production industriels », 12ème Colloque National AIP-PRIMECA, Le Mont-Dore, France, 2011.
- (Pach et al., 2012)** Pach C., Bekrar A., Zbib N., Sallez Y., Trentesaux D., « An effective potential field approach to FMS holonic heterarchical control », Control Engineering Practice, vol. 20, 2012, p. 1293–1309.
- (Pach, 2013)** Pach C., « ORCA : Architecture hybride pour le contrôle de la myopie dans le cadre du pilotage des Systèmes Flexibles de Production », Thèse de doctorat, Université de Valenciennes et du Hainaut-Cambrésis, 2013.
- (Pach et al., 2014)** Pach C., Berger T., Bonte T., Trentesaux D., « ORCA-FMS: a dynamic architecture for the optimized and reactive control of flexible manufacturing scheduling », Computers in Industry, vol. 65, 2014, p.706–720.
- (Pels et al., 1997)** Pels H. J., Wortmann J. C., Zwegers A. J. R., « Flexibility in manufacturing: an architecture point of view », Computer in Industry, vol 33, 1997, p. 271-283.
- (Pannequin et al., 2006)** Pannequin R., Thomas A., « Proposition d'une plateforme d'expérimentation sur le contrôle par le produit des flux de production », MOSIM'6 (Modélisation, optimisation et simulation des systèmes: défis et opportunités), Rabat-Maroc, 2006.
- (Petin, 2007)** Petin J. F., « Méthodes et modèles pour un processus sûr d'automatisation », Habilitation à Diriger des Recherches (HDR), Nancy Université et Centre de Recherche en Automatique de Nancy (CRAN), 2007.
- (Piechocki, 2009)** Piechocki L., « UML : le langage de modélisation objet unifié », Club des développeurs, 2009.
- (Pilone et al., 2008)** Pilone D., Pitman N., Priou D., « UML 2 en concentré », Digit Books, 2008.

- (Proface, 2013)** Pro-face for the best interface, <http://www.proface.fr>, October 2013.
- (Pujo *et al.*, 2002)** Pujo P., Kieffer J. P., « Fondements du pilotage des systèmes de production », Edition Lavoisier, 2002.
- (Pujo *et al.*, 2008)** Pujo P., Ounnar F., « Un modèle holonique et isoarchique pour le pilotage d'un job shop en flux tiré », Conférence Internationale de MODélisation et SIMulation (MOSIM'08), Paris-France, 2008.
- (Pujo *et al.*, 2009a)** Pujo P., Broissin N., Ounnar F., « PROSIS: An isoarchic structure for HMS control », Engineering Applications of Artificial Intelligence, vol 22, 2009, p. 1034–1045.
- (Pujo, 2009b)** Pujo P., « De l'isoarchie pour le pilotage des systèmes de production », Habilitation à diriger des recherches, Université Paul Cézanne Aix-Marseille III, 2009.
- (Quintas *et al.*, 1997)** Quintas A., Leitao P., « A Manufacturing Cell Controller Architecture », Proceedings of Flexible Automation and Intelligent Manufacturing Conference (FAIM'97), Middlesbrough, England, Begell House Inc, ISBN 1-56700-089-4, 1997, p. 483-493.
- (Raileanu, 2011)** Raileanu S., « Proposition d'un modèle générique de pilotage pour un système à flux guidés /Application des concepts holoniques au transport intelligent (FMS/PRT) », Thèse de doctorat, Université Politehnica de Bucarest, Roumanie, 2011.
- (Rochfeld *et al.*, 1983)** Rochfeld A., Tardieu H., « MERISE: An information system design and development methodology », Information & Management, vol. 6, Issue 3, 1983, p. 143 - 159.
- (Roques, 2002)** Roques P., « Les cahiers du programmeur UML - Modéliser un site e-commerce », Edition Eyrolles, 2002.
- (Roques, 2009)** Roques P., « SysML par l'exemple - un langage de modélisation pour systèmes complexes », Edition Eyrolles, 2009.
- (Roques, 2011)** Roques P., « UML 2 par la pratique: Etudes de cas et exercices corrigés », 7^e Edition Eyrolles, 2011.
- (Sallez *et al.*, 2009)** Sallez Y., Berger T., Trentesaux D., « A stigmergic approach for dynamic routing of active products in FMS », Computers in Industry, vol. 60, 2009, p. 204–216.
- (Sallez *et al.*, 2010)** Sallez Y., Berger T., Raileanu S., Chaabane S., Trentesaux D., « Semi-heterarchical control of FMS: From theory to application », Engineering Applications of Artificial Intelligence, vol. 23, 2010, p. 1314-1326.

- (Senehi et al., 1994)** Senehi M. K., Kramer T. R., Ray S. R., Quintero R., Albus J. S., « Hierarchical control architectures from shop level to end effectors », Computer control of flexible manufacturing systems, Editors. Joshi S. B. and Smith J. S., Chapman & Hall, 1994, p. 31-62.
- (Shen et al., 2006)** Shen W., Hao Q., Yoon H. J., Norrie D. H., « Applications of agent-based systems in intelligent manufacturing: An updated review », Advanced Engineering Informatics, vol. 20, 2006, p. 415-431.
- (Shpitalni et al., 2004)** Shpitalni M., Remennik V., « Practical number of paths in reconfigurable manufacturing systems with crossover », The International Journal for Manufacturing Science and Production, vol. 6, n° 1, 2004, p. 9-19.
- (Sillitto, 2013)** Sillitto H. G., « Composable Capability – Principles, Strategies and Methods for Capability Systems Engineering », INCOSE International Symposium, Jacksonville, FL, 24-27 June, 2013.
- (Spicer et al., 2002)** Spicer P., Koren Y., Shpitalni M., « Design Principles for Machining System Configurations », Annals of the CIRP, vol. 51, n° 1, 2002, p. 276-280.
- (Spicer et al., 2005)** Spicer P., Yip-Hoi D., Koren Y., « Scalable reconfigurable equipment design principles », International Journal of Production Research, vol 43, n° 22, 2005, p. 4839-4852.
- (Tamani, 2008)** Tamani K., « Développement d'une méthodologie de pilotage intelligent par régulation de flux adaptée aux systèmes de production », Thèse de doctorat, Université de Savoie, 2008.
- (Techniques de l'ingénieur, 2013)** Techniques de l'ingénieur, <http://www.techniques-ingenieur.fr>, mai 2013.
- (Teunis et al., 1998)** Teunis G., Leitao P., Madden M., « A New Architecture for Flexible Shop Control Systems », Proceedings of Integration in Manufacturing Conference, Gotengorg, Sweden, 1998, p. 699-709.
- (Trentesaux et al., 2000)** Trentesaux D., Pesin P., Tahon C., « Distributed artificial intelligence for SFP scheduling, control and design support », Journal of Intelligent Manufacturing, vol 11, 2000, p. 573–589.
- (Trentesaux, 2002)** Trentesaux D., « Pilotage hétérarchique des systèmes de production », Habilitation à diriger des recherches, Université de Valenciennes et du Hainaut-Cambrésis, 2002.
- (Trentesaux, 2007)** Trentesaux D., « Les systèmes de pilotage hétérarchiques : innovations réelles ou modèles stériles ? », Journal Européen des Systèmes Automatisés, Lavoisier, vol 41, n° 9 - 10, 2007, p. 1165 – 1202.

- (Trentesaux, 2009)** Trentesaux D., « Distributed control of production systems », *Engineering Applications of Artificial Intelligence*, vol. 22, 2009, p. 971–978.
- (Turki, 2008)** Turki S., « Ingénierie système guidée par les modèles: Application du standard IEEE 15288, de l'architecture MDA et du langage SysML à la conception des systèmes mécatroniques », Thèse de doctorat, Université du Sud Toulon-Var, 2008.
- (Ueda *et al.*, 2001)** Ueda K., Hatono I., Fujii N., Vaario J., « Line-less Production System Using Self-organization: A Case Study for BMS », *CIRP Annals-Manufacturing Technology*, vol. 50, n° 1, 2001, p. 319-322.
- (UMLInfra, 2004)** UMLInfra, « Unified Modeling Language (UML) Specification : Infrastructure », version 2.0, 2004.
- (US Air Force, 1981)** US Air Force, ICAM Architecture, « Part II, Functional Modeling Manual (IDEF0) », vol. IV, AFWAL-TR-81-4023, Materials Laboratory, Air Force Wright Aeronautical Laboratories, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio 45433, June 1981.
- (Van Brussel *et al.*, 1998)** Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L., Peeters, P., « Reference architecture for holonic manufacturing systems: PROSA », *Computers in Industry* 37, 255–274
- (White, 1994)** White I., « Rational Rose Essentials: Using the Booch Method », Benjamin/Cummings Publishing Company, 1994.
- (Wiendahl *et al.*, 2004)** Wiendahl H. P., Heger C. L., « Justifying Changeability. A Methodical Approach to Achieving Cost Effectiveness », *The International Journal for Manufacturing Science & Production*, vol. 6, n° (1-2), 2004, p. 33-39.
- (Wiendahl *et al.*, 2007)** Wiendahl H. P., ElMaraghy H. A., Nyhuis P., Zäh M. F., Wiendahl H. H., Duffie N., Brieke M., « Changeable Manufacturing-Classification, Design and Operation », *CIRP Annals-Manufacturing Technology*, vol. 56, n° 2, 2007, p. 783-809.
- (Willard, 2007)** Willard B., « UML for systems engineering », *Computer Standard & interfaces*, vol. 29, 2007, pp. 69-81.
- (Wirsing *et al.*, 2009)** Wirsing M., Knapp A., « A Formal Approach to Object-Oriented Software Engineering », Elsevier, 2009.
- (Zamaï, 1997)** Zamaï E., « Architecture de Surveillance-Commande pour les Systèmes à Événements Discrets Complexes », Thèse de doctorat, Université Paul Sabatier, Toulouse, 1997.
- (Zambrano Rey *et al.*, 2014)** Zambrano Rey G., Bonte T., Prabhu V., Trentesaux D., « Reducing myopic behavior in FMS control: A semi-heterarchical

simulation–optimization approach », *Simulation Modelling Practice and Theory*, vol 46, 2014, p. 53–75.

(Zhu, 2005) Zhu X. W., « Calculating the number of possible system configurations », Engineering Research Center for Reconfigurable Manufacturing Systems Report, 2005.



CONCEPTION ET DEPLOIEMENT DES SYSTEMES DE PRODUCTION RECONFIGURABLES ET AGILES (SPRA)

RESUME: L'industrie est aujourd'hui, comme elle a toujours été, une pierre angulaire de l'économie pour chaque pays développé. Avoir une base solide en entreprises industrielles est très important parce qu'elles poussent et stimulent tous les autres secteurs de l'économie, et offrent également une grande variété d'emplois qui apporte des bonnes conditions de vie dans de nombreux secteurs de la société. L'augmentation de la concurrence mondiale, l'évolution rapide du marché, la nécessité de créer des entreprises stables avec des usines rentables obligent la mise en œuvre d'une démarche globale prenant en compte à la fois les aspects techniques, économiques, logistiques et sociétaux lors de la conception d'un système de production innovant. L'objectif de cette thèse est de contribuer au développement d'un concept innovant de Systèmes de Production Reconfigurables et Agiles (SPRA) permettant de s'adapter rapidement et efficacement aux exigences imposées du marché, des clients, de la technologie des procédés, de l'environnement et de la société afin que l'entreprise soit dynamique, compétitive et rentable. Dans ces travaux de thèse, la proposition d'un modèle générique et la caractérisation de ce nouveau type de système de production ont été décrits en utilisant le langage de modélisation des systèmes complexes (SysML : Systems Modeling Language). Ensuite, nous avons développé un processus de reconfiguration qui représente une démarche à suivre pour concevoir et implanter une nouvelle configuration. De plus, un pilotage opérationnel adapté au SPRA a été introduit. Enfin, quelques travaux développés au cours de cette thèse ont été partiellement déployés sur un démonstrateur industriel au sein de la plate-forme AIP-PRIMECA Auvergne.

Mots-clés : Système de production, Reconfigurabilité, Agilité, SysML, Exigences, Modèle générique, Processus de reconfiguration, Pilotage opérationnel, Ligne d'assemblage.

DESIGN AND DEPLOYMENT OF RECONFIGURABLE AND AGILE MANUFACTURING SYSTEMS (RAMS)

ABSTRACT: Industry is, today as it has always been, a cornerstone of the economy for any developed country. Having a strong manufacturing base is very important because it impels and stimulates all the other sectors of the economy. It provides a wide variety of jobs, which bring higher standards of living to many sectors of society, and builds a strong middle class. Increasing global competition, rapid changes in the marketplace and the need to create stable companies with profitable plants require the implementation of a global approach, taking into account technical, economic, logistic and societal aspects in the design of an innovative manufacturing system. The aim of this dissertation is to contribute to the development of an innovative concept of Reconfigurable and Agile Manufacturing Systems (RAMS) to adapt quickly and effectively to the requirements imposed by markets, customers, technology processes, the environment and society, to ensure that the enterprise is dynamic, competitive and profitable. In this thesis work, the characterization and proposal of a generic model for this new type of manufacturing system have been described using the language of complex systems modeling (SysML: Systems Modeling Language). We have developed a reconfiguration process that represents the approach to follow in the design and implementation of a new configuration. In addition, the operational control of a RAMS has been introduced. Finally, some works developed in this thesis have been partially deployed on an industrial demonstrator within the AIP-PRIMECA Auvergne organisation.

Keywords: Manufacturing system, Reconfigurability, Agility, SysML, Requirements, Generic model, Reconfiguration process, Operational control, Assembly line.